令和二年度卒業論文

外気温湿度と流動人口が呼吸器系感染症の感染者数に及ぼす影響

近畿大学建築学部建築学科

建築環境設備研究室

17-1-191-0281

西田　有輝

目次

第 1 章　序論

1.1　研究背景

2020 年は新型コロナウイルスの感染拡大により、日本でも多くの感染者が発生した。また、代表的な呼吸器感染症として知られているインフルエンザと同様に外気温湿度が低下する冬期にさらなる感染拡大が懸念されている[1)]。

本研究では、日本国内における過去 10 年間のインフルエンザ感染者数と外気温湿度との関連性について検討する。次に、新型コロナウイルス感染症に着目し、日本およびオーストラリアにおける感染者数と外気温湿度との関連性について検討し、インフルエンザを対象とした結果や海外を対象とした結果などと比較する。

一方、新型コロナウイルス感染症については外気温湿度が及ぼす影響とともに人の移動に伴う感染拡大が指摘されている。

本研究では、人の移動に関するデータ（以降、流動人口データと記す）を用いて感染症拡大に与える影響について検討する。

1.1.1　国内の時系列と報道

図 1.1 に時系列と新型コロナウイルス感染者数推移を、表 1.1 に 2020 年の国内の時系列と報道を示す。日本国内では 2020 年 1 月 16 日に武漢に渡航した中国籍の男性から初めて新型コロナウイルスの感染が確認された。その直後に奈良県内で武漢への渡航歴のない人の感染が初めて確認され、瞬く間に日本中に感染が拡大した。4 月 7 日に緊急事態宣言が発令され、5 月 25 日に宣言が解除されるまでは感染者の減少が見られた。しかし宣言が解除された後の 6 月 19 日に全国での移動が緩和されてからは、再び感染が拡大していることが確認できる。このことから人の移動が新型コロナウイルスの感染拡大の要因の一つになっていることが考えられる。

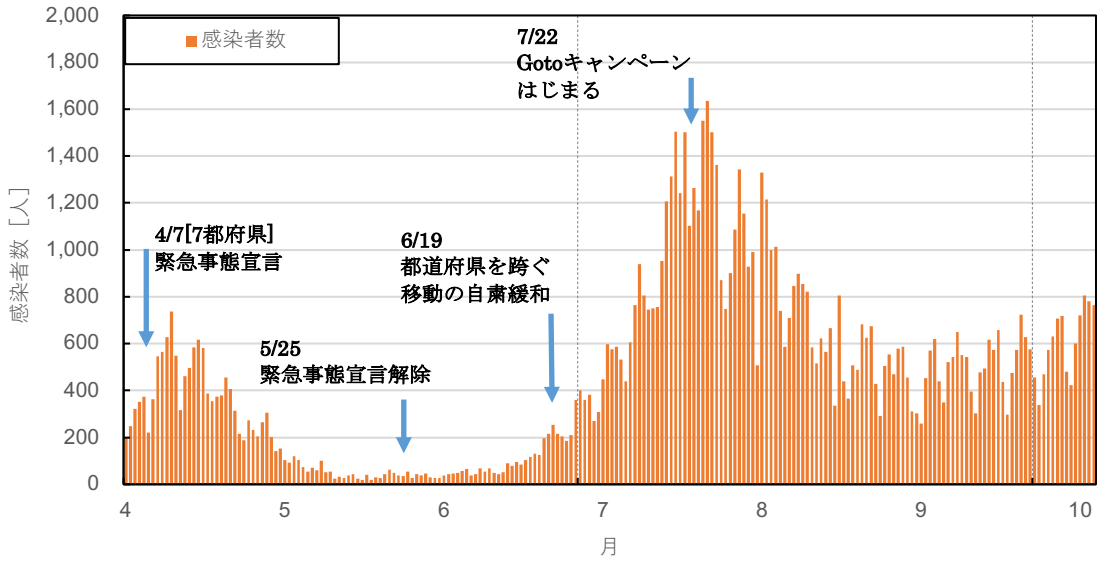


図 1.1 2020 年の時系列と新型コロナウイルス感染者数推移

表 1.1 国内の時系列と報道(2020 年)

1 月 16 日	日本国内で初めて感染確認 武漢に渡航した中国籍の男性
1 月 28 日	武漢へ渡航歴のない人の感染を初確認
2 月 3 日	乗客の感染が確認されたクルーズ船 横浜港に入港
2 月 13 日	国内で初めて感染者死亡 神奈川県に住む 80 代女性
3 月 24 日	東京五輪・パラリンピック 1 年程度延期に
3 月 29 日	志村けんさん死去 新型コロナウイルスによる肺炎で
4 月 7 日	7 都府県に緊急事態宣言 「人の接触 最低 7 割極力 8 割削減を」
4 月 11 日	国内の感染者 1 日の人数としてはこれまでで最多の 700 人超
5 月 7 日	国内の感染者 1 日の人数が 100 人下回る
5 月 25 日	緊急事態の解除宣言 約 1 か月半ぶりに全国で解除
6 月 2 日	初の「東京アラート」 都民に警戒呼びかけ
6 月 19 日	都道府県またぐ移動の自粛要請 全国で緩和
7 月 3 日	国内の 1 日の感染者 2 か月ぶりに 200 人超える
7 月 22 日	「GoTo トラベル」キャンペーン始まる
8 月 17 日	4-6 月期 GDP 年率-27.8%
8 月 20 日	対策分科会 尾身会長「流行はピークに達したとみられる」
9 月 10 日	全国で 713 人感染 東京都は 1 週間ぶりに 200 人超
9 月 26 日	大阪府で新たに 66 人が感染 うち 47 人の経路わからず
10 月 2 日	東京で新たに 203 人感染 2 日連続の 200 人超
10 月 23 日	「年末年始休暇の分散取得を」 業界団体に政府呼びかけ
11 月 7 日	北海道 警戒ステージ「3」に ススキノで営業時間短縮など要請
11 月 18 日	日本医師会の中川会長 「Go To トラベル」と感染「間違いなく十分に関与」
12 月 3 日	大阪府が「医療非常事態宣言」 重症患者の急増で 不要不急の外出自粛も要請
12 月 15 日	Goto トラベル全国一時停止へ 地域限定の対応から方針転換 政府

1.1.2　海外の時系列と報道

表 1.2 に 2020 年の海外の時系列と報道を示す。中国の武漢市で肺炎にかかる人が急増した 1 月 6 日の段階では、まだ原因不明の肺炎とされ新型コロナウイルスと断定はされていなかった。しかし翌月の 2 月 20 日には中国での死者が 2118 人にも達し、3 月 16 日にはドイツで感染者が 4 千人を超え出入国の禁止をしたことから数ヶ月で急速に世界へウイルスが拡大して行ったことがわかる。

時系列全体からドイツやフランス､イギリスなどの寒冷地であるヨーロッパでの報道が多いことも確認できる。

表 1.2 海外の時系列と報道

1月 6 日	中国 武漢で原因不明の肺炎 厚労省が注意喚起
1月 30 日	ＷＨＯ「国際的な緊急事態」を宣言
2月 2 日	武漢の新型肺炎病院、10 日で完成　東京ドームの約半分
2月 20 日	中国、新型肺炎の死者 2118 人に　航空事業の不振深刻
3月 4 日	新型コロナ「インフルほど感染力高くない」　WHO 見解
3月 16 日	ドイツが出入国禁止へ　感染 4 千人超、慎重姿勢から転換
4月 7 日	コロナ感染、世界で 135 万人超す　死者は 7 万 6 千人
4月 17 日	韓国で「再陽性」163 人　隔離解除から平均 13.5 日
5月 14 日	コロナ感染、アフリカの全ての国で確認　最後はレソト
5月 28 日	米、コロナ死者 10 万人超　感染者 169 万人で世界最多
6月 8 日	世界の感染者 24 時間で最多の 13 万 6000 人
6月 28 日	世界の感染者 1000 万人超える
7月 13 日	WHO 「多くの国が誤った方向に」 事態悪化を警告
7月 18 日	世界の死者 60 万人超える
8月 10 日	アメリカの感染者数が 500 万人を超える
8月 15 日	ヨーロッパで感染再拡大受けた措置相次ぐ
9月 5 日	WHO 「新型コロナのワクチン 分配開始は来年中頃の見通し」。
9月 9 日	世界の製薬会社など 9 社が新型コロナワクチン開発で“安全最優先”を宣言
10 月 2 日	トランプ大統領が新型コロナウイルスに感染
10 月 14 日	フランスが 3 か月ぶりに非常事態を宣言　ヨーロッパで感染再拡大
11 月 10 日	ファイザーがワクチン「90%超の予防効果」と暫定結果発表
11 月 24 日	米厚生長官 コロナワクチン 来月 10 日以降供給開始の見通し示す
12 月 8 日	イギリスで新型コロナウイルスのワクチン接種が始まる
12 月 20 日	変異ウイルス拡大 英からの旅客機受け入れ停止 欧州諸国が警戒

1,1,3　既往研究

本研究を進めるにあたり、参考としている研究事例をこの節で述べる。参考文献 1 ではブラジルの気温と新型コロナウイルス感染者数の関係を分析し、ブラジルの年間平均気温の 16.8℃～27.4℃の範囲で気温と新型コロナウイルス感染者数の間には負の線形関係があることを示している。また非線形回帰モデルの解析では、予測決定係数が 0.81 という高い分析制度を出している。しかしこの研究事例では気温との関係性のみ分析されており、湿度は分析対象とされていない。

よって本研究では気温に加えて湿度も分析対象とし、日本での外気温湿度と新型コロナウイルスの関係性を検討する。

1.1.4　海外の同時流行の懸念

日本で新型コロナウイルスの感染が拡大した時期は 3 月からで、日本の季節は冬ではなかった。従来の季節性インフルエンザが流行する時期は外気温湿度が低下する冬季なので、新型コロナウイルスとインフルエンザウイルスの 2 つが同時流行する危険性は少なかった。

しかし南半球のオーストラリアの気候は図 1.2 に示すように、7 月が一年で最も気温が低くなりこの時期がオーストラリアでの冬とされている。そして 7 月は世界中で新型コロナウイルスの感染が拡大している時期であり、オーストラリアの冬における同時流行懸念が懸念されている。

よって、本研究ではオーストラリアでの気温とインフルエンザウイルス及び新型コロナウイルスの関係性を検討する。

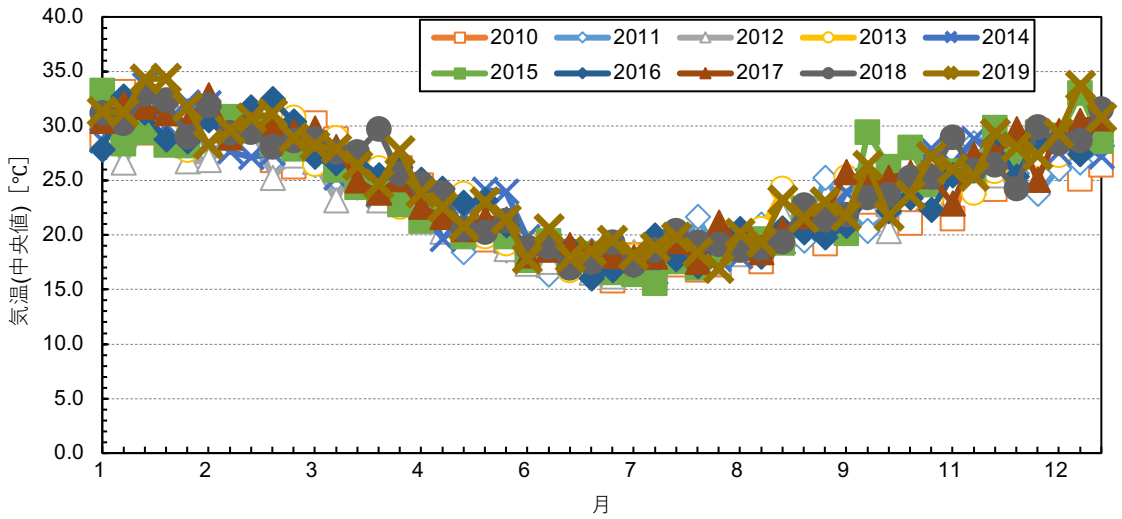


図 1.2 オーストラリアの 10 年間の気温の傾向

1.2　研究目的

本研究では外気温湿度や人の移動と感染症との関係性を検討し、気象条件の変化や人の移動の増減が新型コロナウイルスの感染拡大の要因になっているのかを明らかにする。

1.3　論文フロー図

卒業論文タイトル
「外気温湿度と流動人口が呼吸器系感染症の感染者数に及ぼす影響」

第 1 章　序論 研究背景 研究目的

第 2 章　研究方法 研究概要 分析手法

第 3 章　各種データ概要 気象データ 感染症データ 流動人口データ データ収集方法

第 4 章　外気温湿度と感染症の関係性を検討 感染症ごとの感染者数比較 外気温湿度と感染者数の関係 海外の気温と感染者数の関係

第 5 章　流動人口と新型コロナウイルスの関係性を検討 都道府県ごとの比較 全国を視点にした流動人口と感染者数の関係

第 6 章　総括 全体のまとめ

第 2 章　研究方法

2.1　研究概要

分析の概要を表 2.1 に、感染症ごとの分析期間を表 2.2 に示す。データの出典は表 2.3 を参照されたい。

オーストラリアは湿度のデータを収集することができなかったため、気温のみで分析を行う。新型コロナウイルスの分析期間に関しては、外気温湿度との分析では 4 月～10 月、流動人口との分析では 6 月～11 月の期間で分析を行なっている。

表 2.1　分析概要

	分析項目	感染症
日本	外気温湿度	インフルエンザウイルス感染者数
		新型コロナウイルス感染者数
	流動人口	新型コロナウイルス感染者数
オーストラリア	気温	インフルエンザウイルス感染者数
		新型コロナウイルス感染者数

表 2.2　分析期間

インフルエンザウイルス	2010～2020
新型コロナウイルス	2020

表 2.3 データ出典

	日本	オーストラリア
気温	気象庁	Australian Govemment
湿度		
インフルエンザウイルス	国立感染症研究所	WHO
新型コロナウイルス	Esri ジャパン	内閣官房
流動人口	内閣官房	

2.2　分析手法

以下に示す線型回帰モデルと二次の非線形回帰モデルを用いて、相関係数(以下:r)決定係数(以下:r2)をそれぞれ(1)式と(2)式から算出する。記号表を 2.4 に示す。r 値と r^2 及び、二次曲線近似式より得られた回帰係数a_0～a_2を用いて解析結果を比較検討する。

r 値と r^2 値の詳しい計算方法については付 1 に、計算式全体のソースコードについては付 2 に示す。

$$y = \frac{S_{xy}}{S_x^2}x + \bar{y} - \frac{S_{xy}}{S_x^2}\bar{x} \tag{1}$$

$$X^t X A = X^t Y \tag{2}$$

表 2.4 記号表

S_{xy}:	共分散
S_x^2:	説明変数xの分散
X:	Σx_i^0～Σx_i^4までの範囲の(3×3)の行列
X^t:	Xの転置行列
Y:	$\Sigma x_i^0 y_i$～$\Sigma x_i^2 y_i$までの範囲の(3×1)の行列
A:	二次曲線の係数a_0～a_2までの(3×1)の行列

第 3 章　各種データ概要

3.1　気象データ

気象データを公開している各国の Web サイト名については表 3.1 に示す。気象データはその国ごとの公的な機関がオープンデータを公開している場合が多い。研究に使用したサイトは日本の気象庁とオーストラリアの Australian Govemment である。

日本と海外の気象データには、項目数の違いに差がある。日本の気象データは、気温や相対湿度に加えて、風速風向や日射量に加えて積雪量などの様々な項目が公開されている。海外の場合は気温と降水量だけが公開されている場合が多い。

気象庁から収集した項目は外気温と相対湿度であるが、分析では扱わずに相対湿度から絶対湿度を算出している。絶対湿度の計算に使用した式は以下に示す。計算式の記号表は表 3.2 に示す。

オーストラリアの Australian Govemment からは気温のみを収集している。地域ごとの観測であり最高気温と最低気温のみの項目となっている。オーストラリア全土の気温の傾向で分析するために、表 3.3 に示す主要 5 都市の最高気温最低気温それぞれの平均の中央値を算出して、この加工したデータを分析に使用している。

表 3.1 気象データセット

日本	気象庁
オーストラリア	Australian Govemment
ブラジル	Instituto Nacional de Meteorologia

$$e = 6.1078 \cdot 10^{\frac{7.5t}{t+237.3}} \tag{3}$$

$$a = \frac{217 \cdot e}{t+273.15} \tag{4}$$

$$VH = a \cdot \frac{RH}{100} \tag{5}$$

表 3.2 記号表

t:	気温[℃]
e:	飽和水蒸気圧[hPa]
a:	飽和水蒸気量[g/m³]
RH:	相対湿度[%]
VH:	絶対湿度[g/kg]

表 3.3 オーストラリア主要 5 都市

オーストラリア主要 5 都市	キャンベラ
	シドニー
	メルボルン
	アデレード
	ブリズベン

表 3.4 気象データ URL

気象庁	https://www.jma.go.jp/jma/index.html
Australian Govemment	http://www.bom.gov.au/?ref=logo
Instituto Nacional de Meteorologia	https://portal.inmet.gov.br
EOLdata	https://data.eol.ucar.edu
WMO	https://public.wmo.int/en

3.2　感染症データ

インフルエンザウイルスに関するデータを公開しているWebサイト名を表3.5に、新型コロナウイルスに関するデータを公開しているWebサイト名を表3.6に示す。

インフルエンザウイルス感染者数データは、日本では国立感染症研究所が公開している。過去数十年の地域ごとの感染者数データをこのサイトから収集することができる。国立感染症研究所は全国の定点ごとに感染者数を主計しており、その全定点の合計値も公開している。国単位での感染者数は、WHOが世界各国のデータを公開している。オーストラリアでのインフルエンザ感染者数の分析に関してはWHOから収集している。WHOは週ごとに検体数あたりの感染者数を公開しており、週ごとに検体数は異なる。よって感染者数を検体数で割って100倍し、インフルエンザウイルス陽性率を算出して分析に使用している。

新型コロナウイルス感染者数データは、日本ではEsriジャパンが多くの項目を公開している。分析にあたってこのサイトから新規感染者数と地域ごとの人口を収集し、新たに人口10万人あたり新規感染者数を算出している。日本全国の感染者数を比較する際は、人口の差が分析に考慮されない人口10万人あたり新規感染者数データを使用している。世界各国の新型コロナウイルス感染者数は内閣官房とWHOが公開しており、オーストラリアの感染者数データは内閣官房から収集している。内閣官房の感染者数は累計の感染者数であるため、新規感染者数に加工して分析に使用している。

表3.5 インフルエンザウイルスデータセット

日本	国立感染症研究所
世界各国	WHO

表3.6 新型コロナウイルスデータセット

日本	Esriジャパン
世界各国	内閣官房
	WHO

表3.7 感染症データURL

国立感染症研究所	https://www.niid.go.jp/niid/ja/
WHO	https://www.who.int
Esriジャパン	https://www.esrij.com
内閣官房	https://www.cas.go.jp
NEWYORKSTATE	https://www.ny.gov

3.3　人の移動に関するデータ

流動人口データは、表 3.8 に示す 2 つのサイトが公開している。新型コロナウイルスとの関係性を分析する際に使用した流動人口データは、内閣官房から収集している。

内閣官房の流動人口は、それぞれの都道府県の主要駅(東京都なら東京駅)での人の流れを解析したデータを公開している。解析した時間帯は 15 時である。項目は 3 つ公開しており、前日との比較と感染拡大以前(1 月～2 月)との比較、そして宣言前(4 月 7 日)との比較したデータを公開している。感染拡大以前と比較した人の移動量と宣言前と比較した人の移動量の増減値はほぼ一致しており、分析では宣言前比較のみを使用している。

表 3.8 流動人口データセット表

流動人口データセット	内閣官房
	Agoop

表 3.9 流動人口データ URL

内閣官房	https://www.cas.go.jp
Agoop	https://www.agoop.co.jp
RESAS	https://resas.go.jp
MarketDiscovery	https://www.giken.co.jp

3.4　データ収集方法

データを収集する際に使用したツールはプログラミング言語の Python であり、その際に使用した Web スクレイピング用のライブラリを表 3.10 に示す。

request ライブラリは Web サーバに直接 HTML の提供を求めるライブラリであり、Web サイトを読み込んだ際に生じる Javascript の読み込みが発生する前の HTML が Web サーバから提供される。このライブラリは Javascript の読み込みを考慮できないという欠点があるが非常に高速で HTML を解析できることから、オープンデータを Javascript で処理しない Web サイトの場合はこのライブラリを使用している。

Selenium は Web サイトをプログラムで自動操作をするライブラリである。HTML の読み込みが request よりは遅いが Javascript の読み込みを待つことができるという利点がある。オープンデータが Javascript によって処理される場合はこのライブラリを使用している。

表 3.10 Web スクレイピング用ライブラリ(Python)

Web スクレイピング用ライブラリ	request
	Selenium

第 4 章　外気温湿度と感染症の関係性を検討

4.1　感染症ごとの感染者数を比較

インフルエンザウイルスと新型コロナウイルスの感染者数を比較したものを図 4.1 に示す。インフルエンザウイルス感染者数の集計期間は一年間、新型コロナウイルス感染者数の集計期間は 4 月～10 月となっている。

2019 年のインフルエンザウイルス感染者数と 2020 年の新型コロナウイルス感染者数の感染者数を比較すると、100 万人以上の差が見られる。そして、2019 年と 2020 年のインフルエンザウイルス感染者数を比較すると、2020 年は約 90 万人感染者数が少ないことも確認できる。

2020 年のインフルエンザ感染者数が減少している要因として、新型コロナウイルスに対する感染対策が影響しているのではないかと考える。

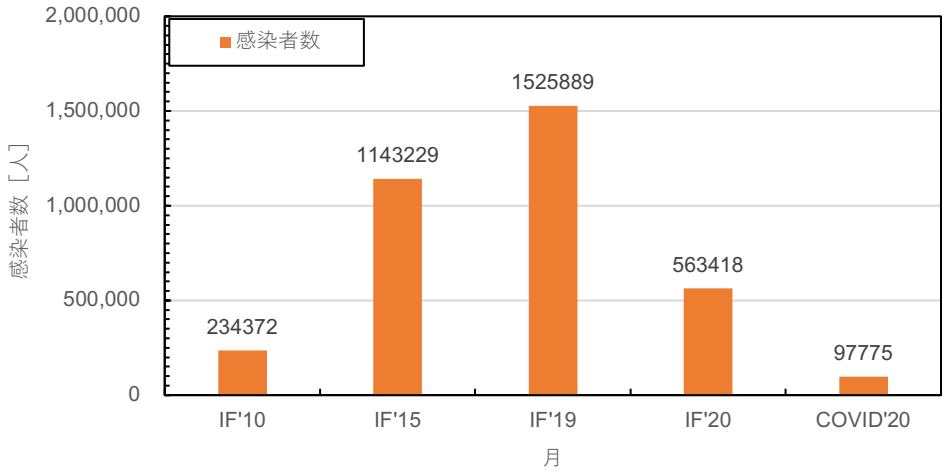


図 4.1 感染症ごとの感染者数比較

4.2　外気温湿度と感染者数の関係

4.2.1　インフルエンザウイルス感染者数との関係

都道府県ごとの年平均気温を図 4.2 に、年平均絶対湿度を図 4.3 に示す。これらの図は 10 年間の推移を表しており、気温と絶対湿度ともに北海道から沖縄にかけて数値が上がる傾向にあることがわかる。図 4.4 の都道府県ごとの感染者数を比較した図からは首都圏や関西圏などの人口が集中している地域に感染者が集中していることがわかり、図 4.5 の週ごとの感染者数推移では第一週から第九週にかけて感染者数が集中する傾向にあることが確認できる。

図 4.6 から図 4.14 にかけては、北海道，東京，愛知，大阪の主要 4 都市での 10 年間の外気温湿度とインフルエンザウイルス感染者数の関係性を図に示す。2.2 節に示した回帰式を用いて得られた結果を表 4.1 から表 4.20 に示す。外気温湿度との関係性を示した図からは、どの年も外気温が 10℃以下絶対湿度が 5g/kg 付近で感染者数が集中する傾向にあることがわかる。解析結果を示した表からは、r 値が-0.60 付近で r^2 が 0.50 付近と比較的高い負の関係性を示している。2015 年の大阪の外気温と感染者数の回帰係数を例にすると、気温が 1℃減少するごとに感染者数が 11337 人発生するという予測値になる。

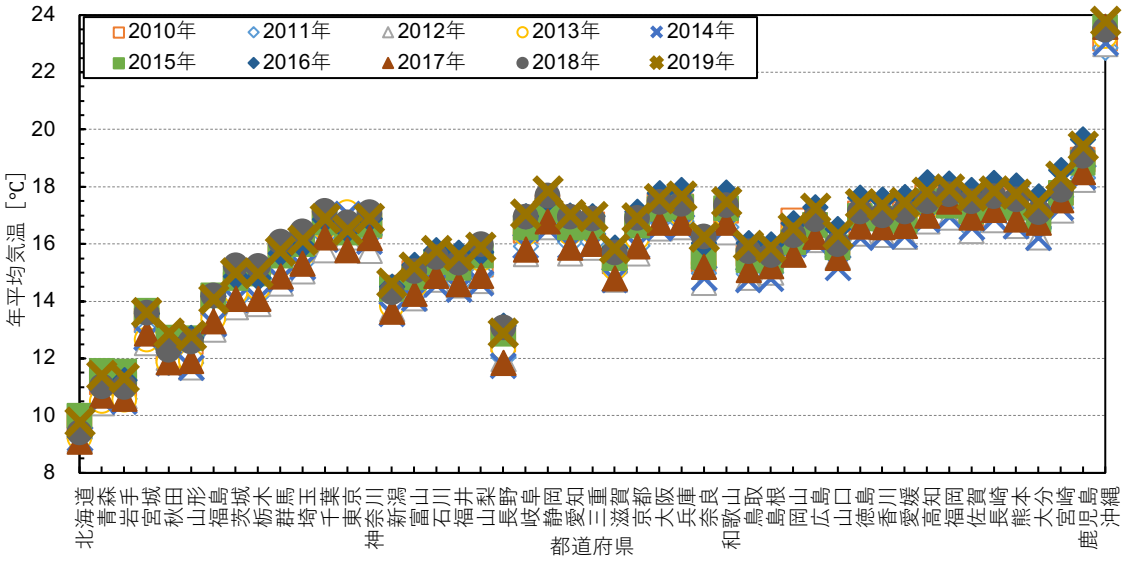


図 4.2 都道府県ごとの年平均気温(10 年間)

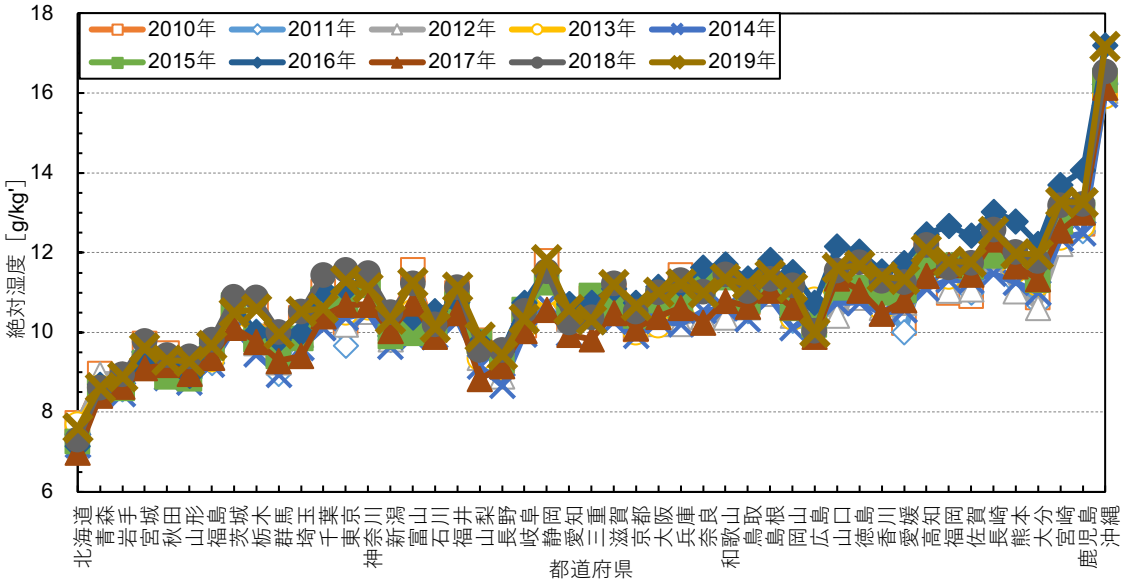


図 4.3 都道府県ごとの年平均絶対湿度(10 年間)

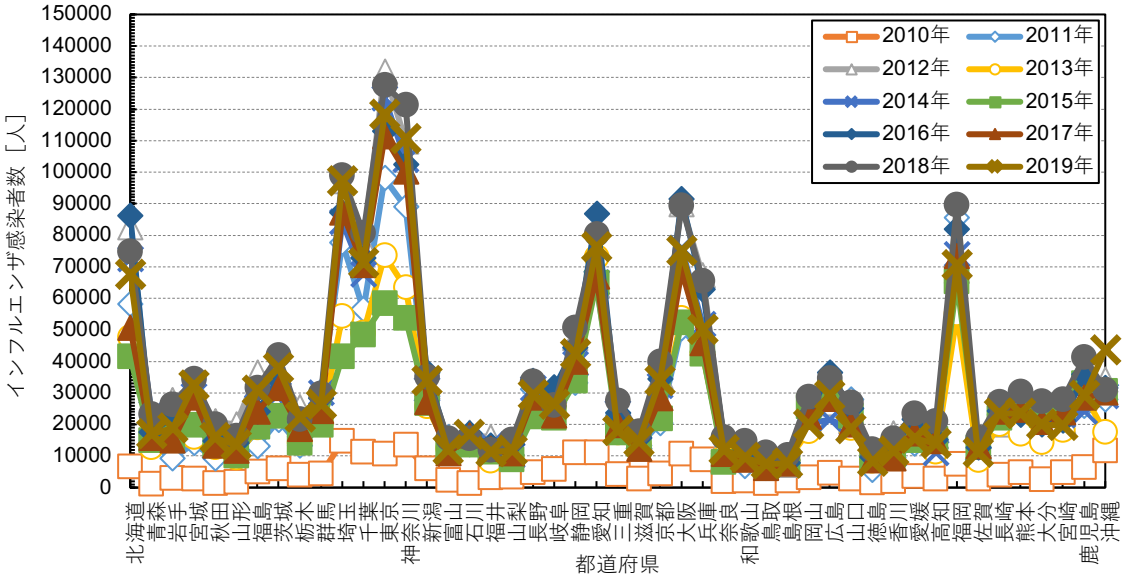


図 4.4 都道府県ごとの感染者数比較(10 年間)

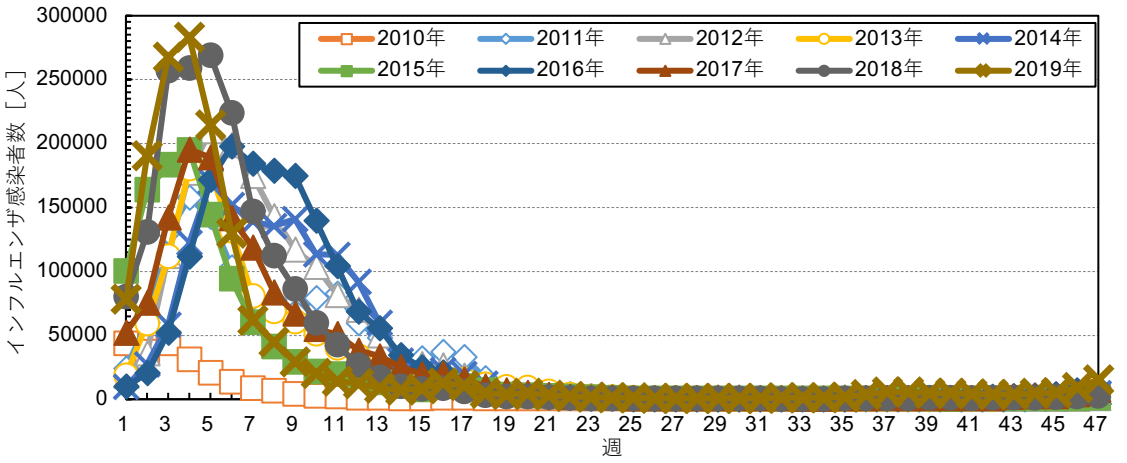
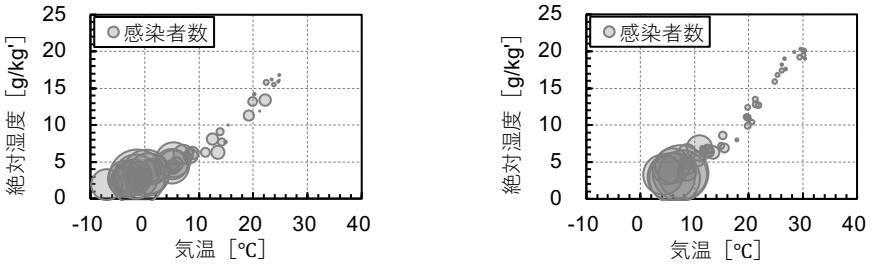
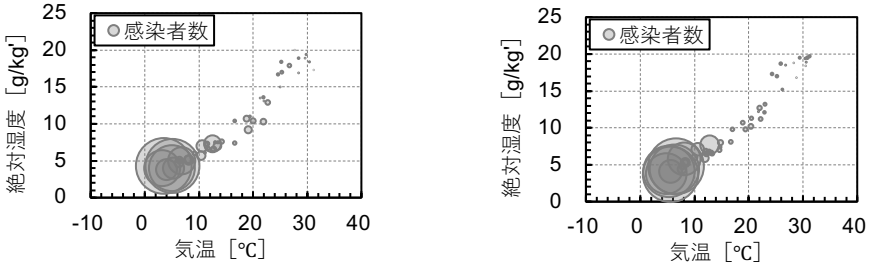


図 4.5 週ごとの感染者数推移(全国:10 年間)

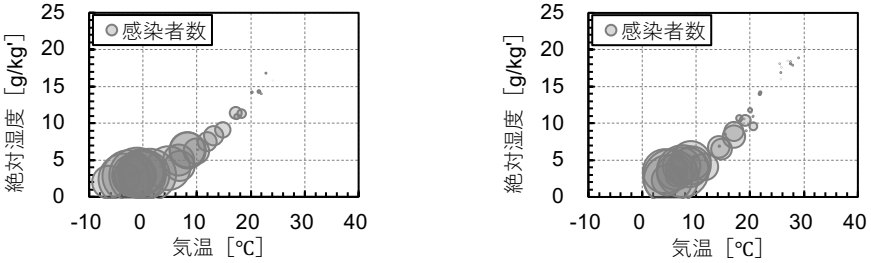


北海道　　東京

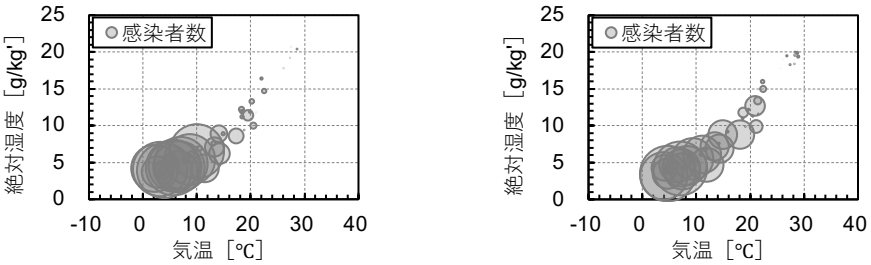


愛知　　大阪

図 4.6 外気温湿度とインフルエンザ感染者数の関係(2010 年)



北海道　　東京



愛知　　大阪

図 4.6 外気温湿度とインフルエンザ感染者数の関係(2011 年)

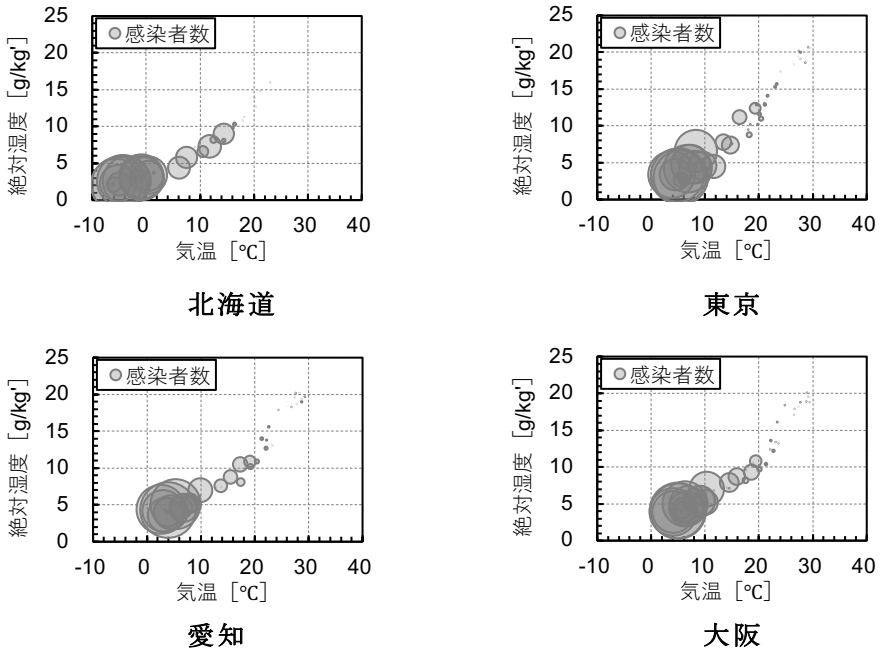


図 4.7 外気温湿度とインフルエンザ感染者数の関係(2012 年)

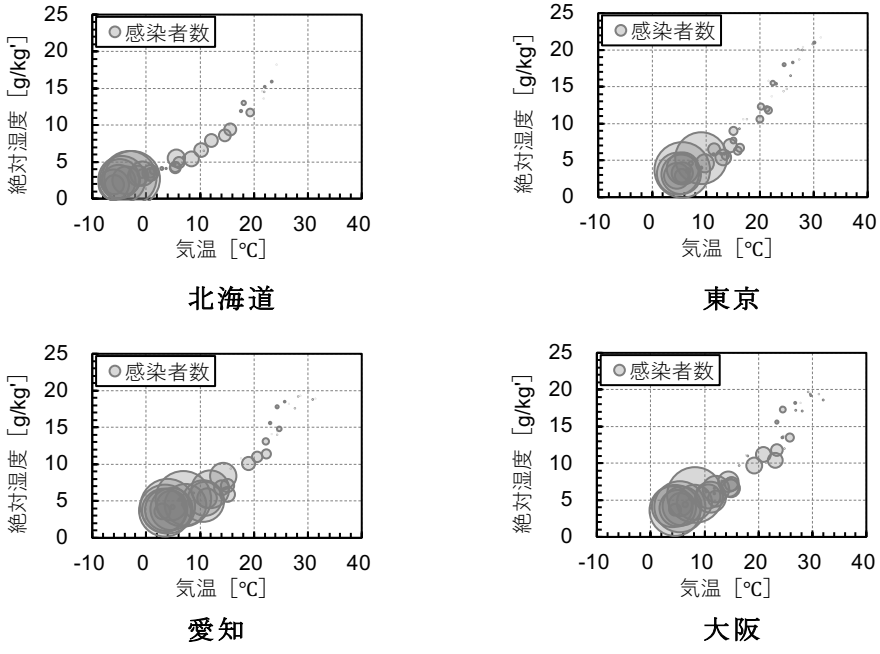


図 4.8 外気温湿度とインフルエンザ感染者数の関係(2013 年)

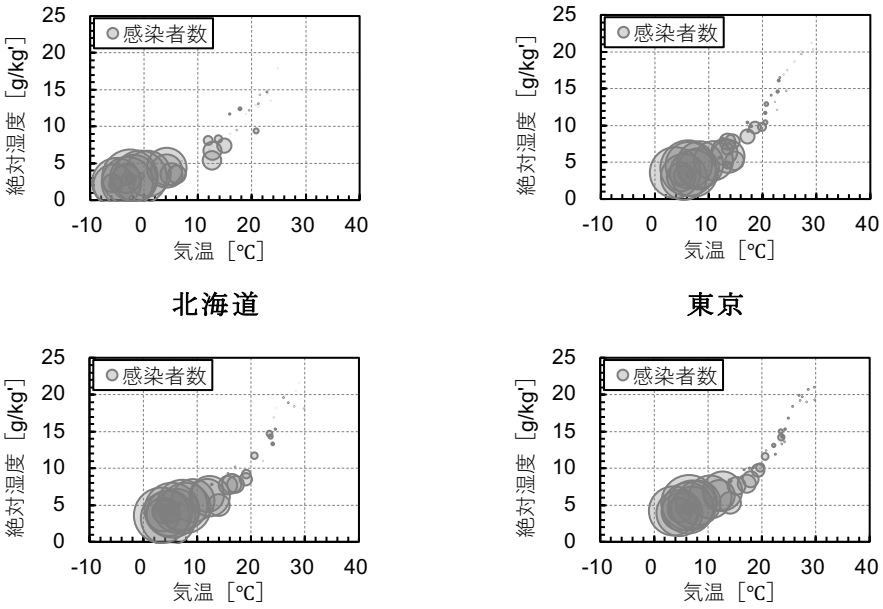


図 4.9 外気温湿度とインフルエンザ感染者数の関係(2014 年)

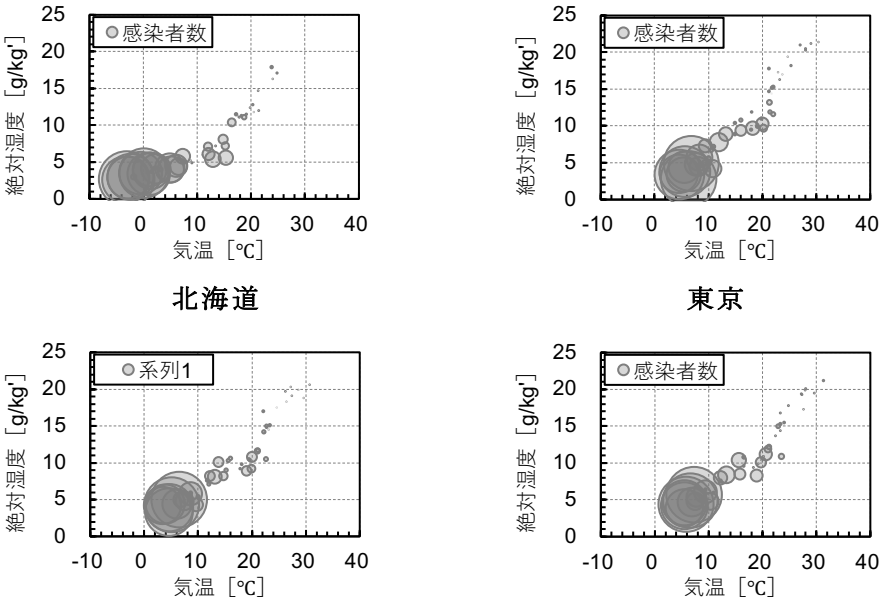


図 4.10 外気温湿度とインフルエンザ感染者数の関係(2015 年)

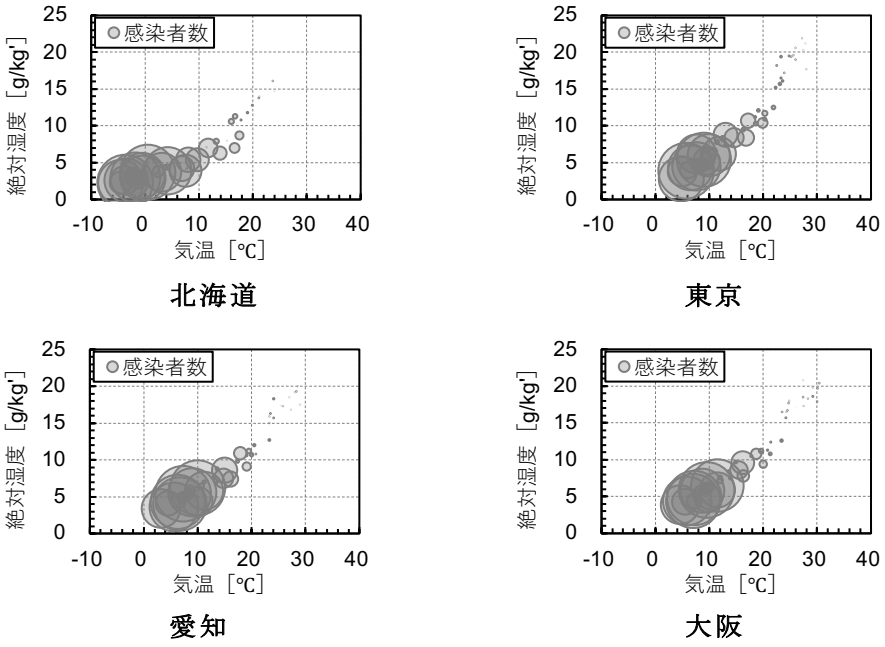


図 4.11 外気温湿度とインフルエンザ感染者数の関係(2016 年)

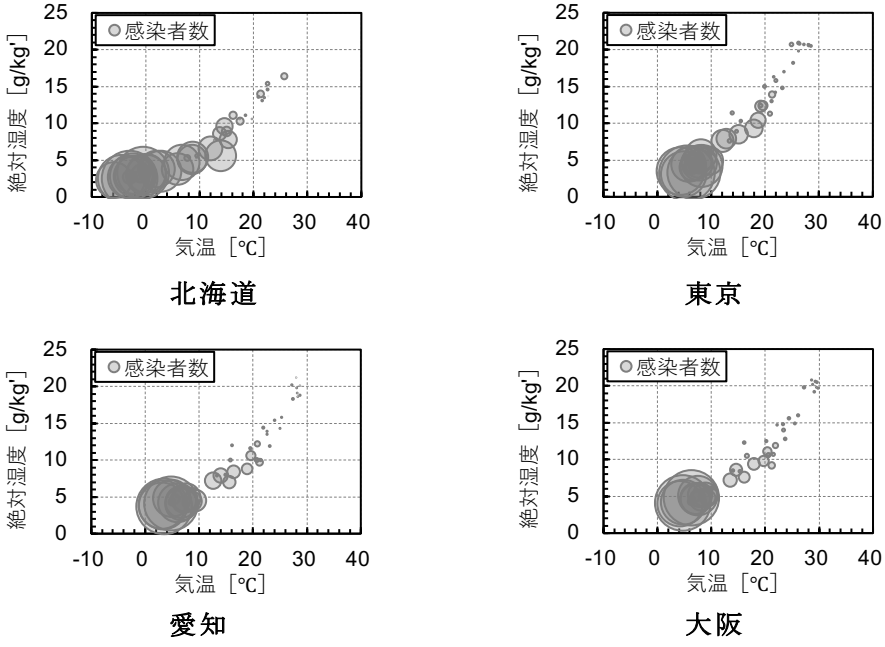


図 4.12 外気温湿度とインフルエンザ感染者数の関係(2017 年)

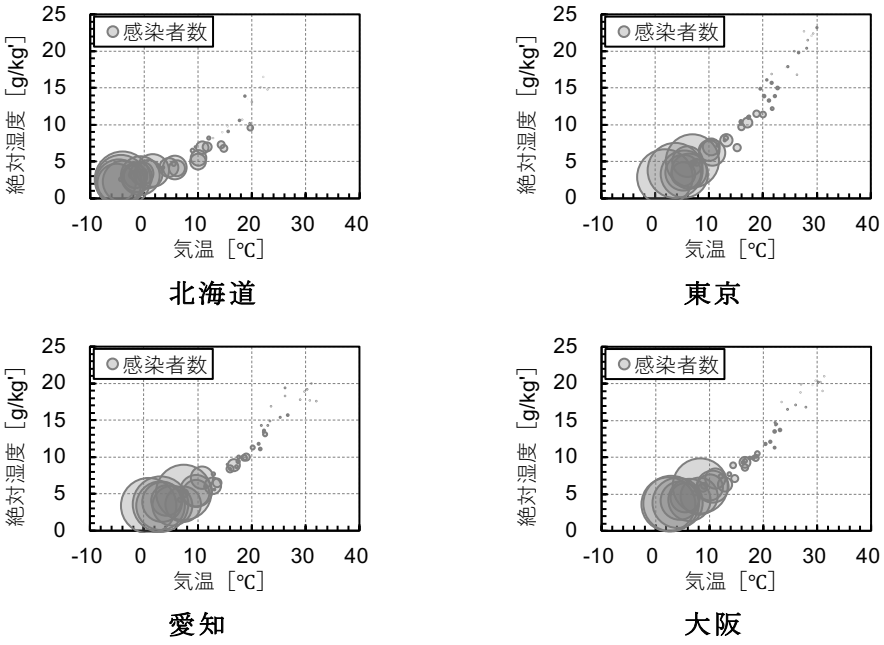


図 4.13 外気温湿度とインフルエンザ感染者数の関係(2018 年)

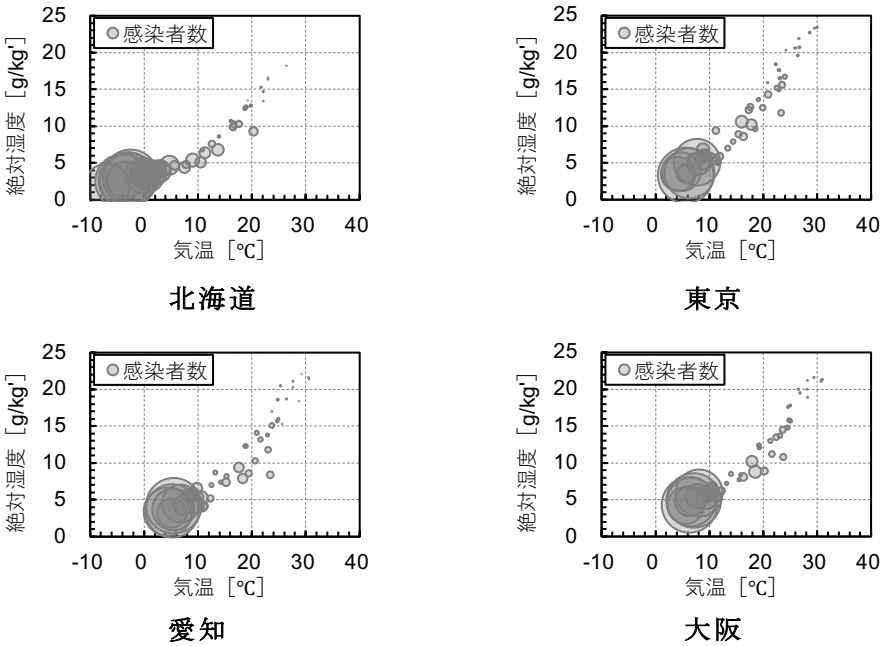


図 4.14 外気温湿度とインフルエンザ感染者数の関係(2019 年)

表 4.1　外気温とインフルエンザウイルス感染者数の解析結果(2010 年)

	北海道	東京	愛知	大阪
r	-0.67	-0.55	-0.50	-0.52
r^2	0.49	0.50	0.46	0.51
a_0	272.13	1645.17	1808.45	1958.15
a_1	-23.62	-159.44	-189.30	-196.47
a_2	0.50	3.60	4.48	4.50

表 4.2 絶対湿度とインフルエンザウイルス感染者数の解析結果(2010 年)

	北海道	東京	愛知	大阪
r	-0.59	-0.51	-0.40	-0.43
r^2	0.46	0.55	0.33	0.40
a_0	537.12	1627.18	1957.95	1931.75
a_1	-91.45	-270.02	-322.84	-317.93
a_2	3.65	9.80	11.84	11.59

表 4.3　外気温とインフルエンザウイルス感染者数の解析結果(2011 年)

	北海道	東京	愛知	大阪
r	-0.62	-0.65	-0.65	-0.63
r^2	0.39	0.51	0.47	0.42
a_0	2236.86	10587.45	6291.83	4201.61
a_1	-140.39	-906.45	-493.66	-295.38
a_2	1.52	18.99	9.46	5.14

表 4.4 絶対湿度とインフルエンザウイルス感染者数の解析結果(2011 年)

	北海道	東京	愛知	大阪
r	-0.62	-0.60	-0.58	-0.60
r^2	0.39	0.49	0.49	0.47
a_0	2236.86	9511.15	8335.30	5017.37
a_1	-140.39	-1417.86	-1174.78	-672.67
a_2	1.52	49.27	38.81	21.46

表 4.5　外気温とインフルエンザウイルス感染者数の解析結果(2012 年)

	北海道	東京	愛知	大阪
r	-0.60	-0.59	-0.58	-0.61
r^2	0.42	0.47	0.46	0.49
a_0	3128.18	15532.64	6800.00	10231.13
a_1	-332.19	-1459.86	-684.78	-945.56
a_2	8.56	32.76	16.22	20.91

表 4.6　絶対湿度とインフルエンザウイルス感染者数の解析結果(2012 年)

	北海道	東京	愛知	大阪
r	-0.54	-0.51	-0.48	-0.53
r^2	0.41	0.35	0.34	0.42
a_0	7265.21	12436.49	7778.58	11290.61
a_1	-1304.42	-1736.53	-1151.29	-1696.73
a_2	53.87	56.70	39.19	58.66

表 4.7　外気温とインフルエンザウイルス感染者数の解析結果(2013 年)

	北海道	東京	愛知	大阪
r	-0.57	-0.51	-0.61	-0.58
r^2	0.45	0.38	0.44	0.42
a_0	2003.42	10738.54	6542.55	5661.01
a_1	298.71	-1004.17	-535.07	-480.34
a_2	5.14	22.35	10.76	9.98

表 4.8 絶対湿度とインフルエンザウイルス感染者数の解析結果(2013 年)

	北海道	東京	愛知	大阪
r	-0.60	-0.42	-0.56	-0.51
r^2	0.47	0.27	0.42	0.36
a_0	5017.37	8144.79	8102.73	6026.50
a_1	-672.67	-1185.24	-1173.32	-872.88
a_2	21.46	39.55	39.94	29.85

表 4.9　外気温とインフルエンザウイルス感染者数の解析結果(2014 年)

	北海道	東京	愛知	大阪
r	-0.57	-0.69	-0.61	-0.64
r^2	0.35	0.61	0.40	0.46
a_0	2885.46	16884.96	5141.96	7122.37
a_1	-266.00	-1475.53	-371.04	-533.89
a_2	5.95	31.36	6.52	10.63

表 4.10　絶対湿度とインフルエンザウイルス感染者数の解析結果(2014 年)

	北海道	東京	愛知	大阪
r	-0.49	-0.59	-0.56	-0.56
r^2	0.32	0.51	0.46	0.42
a_0	6088.48	14373.09	7004.57	8161.73
a_1	-1097.13	-2055.19	-1000.69	-1106.53
a_2	46.28	68.35	33.21	35.41

表 4.11　外気温とインフルエンザウイルス感染者数の解析結果(2015 年)

	北海道	東京	愛知	大阪
r	-0.67	-0.62	-0.61	-0.62
r^2	0.62	0.57	0.58	0.58
a_0	2354.77	9359.95	10278.31	9165.36
a_1	-322.55	-936.64	-1034.08	-889.51
a_2	10.21	22.34	24.51	20.51

表 4.12 絶対湿度とインフルエンザウイルス感染者数の解析結果(2015 年)

	北海道	東京	愛知	大阪
r	-0.52	-0.54	-0.51	-0.51
r^2	0.46	0.51	0.48	0.45
a_0	4582.13	8233.67	10726.54	8632.60
a_1	-877.56	-1206.97	-1625.39	-1259.69
a_2	37.77	40.41	56.30	45.32

表 4.13　外気温とインフルエンザウイルス感染者数の解析結果(2016 年)

	北海道	東京	愛知	大阪
r	-0.60	-0.61	-0.58	-0.58
r^2	0.38	0.46	0.39	0.42
a_0	3485.03	15055.42	9608.77	11874.28
a_1	-300.54	-1395.61	-790.62	-993.06
a_2	6.27	31.58	15.97	20.25

表 4.14　絶対湿度とインフルエンザウイルス感染者数の解析結果(2016 年)

	北海道	東京	愛知	大阪
r	-0.53	-0.55	-0.53	-0.54
r^2	0.37	0.45	0.41	0.43
a_0	6943.54	13942.89	11272.16	13390.62
a_1	-1229.16	-1928.75	-1643.10	-1914.10
a_2	50.64	62.15	56.08	64.05

表 4.15　外気温とインフルエンザウイルス感染者数の解析結果(2017 年)

	北海道	東京	愛知	大阪
r	-0.66	-0.62	-0.62	-0.57
r^2	0.45	0.51	0.55	0.50
a_0	1939.29	13396.03	8102.25	9869.22
a_1	-145.55	-1290.33	-831.99	-978.53
a_2	2.59	30.07	19.99	22.70

表 4.16 絶対湿度とインフルエンザウイルス感染者数の解析結果(2017 年)

	北海道	東京	愛知	大阪
r	-0.61	-0.59	-0.52	-0.48
r^2	0.47	0.53	0.44	0.36
a_0	3960.49	12389.36	8232.71	8739.86
a_1	-696.81	-1750.58	-1242.04	-1268.54
a_2	29.84	57.21	42.72	42.27

表 4.17　外気温とインフルエンザウイルス感染者数の解析結果(2018 年)

	北海道	東京	愛知	大阪
r	-0.66	-0.63	-0.67	-0.67
r^2	0.58	0.64	0.67	0.70
a_0	3463.98	20297.46	10674.40	13570.29
a_1	-500.79	-2036.52	-1004.55	-1267.81
a_2	16.49	47.93	22.19	27.88

表 4.18　絶対湿度とインフルエンザウイルス感染者数の解析結果(2018 年)

	北海道	東京	愛知	大阪
r	-0.56	-0.54	-0.56	-0.56
r^2	0.51	0.50	0.52	0.58
a_0	8651.59	18173.45	11859.87	15291.93
a_1	-1758.60	-2506.31	-1867.20	-2291.83
a_2	80.73	78.70	67.43	78.14

表 4.19　外気温とインフルエンザウイルス感染者数の解析結果(2019 年)

	北海道	東京	愛知	大阪
r	-0.70	-0.53	-0.56	-0.55
r^2	0.70	0.46	0.57	0.50
a_0	3215.11	20464.31	14428.20	14506.53
a_1	-494.51	-2095.15	-1513.08	-1443.82
a_2	16.29	50.46	36.50	33/63

表 4.20 絶対湿度とインフルエンザウイルス感染者数の解析結果(2019 年)

	北海道	東京	愛知	大阪
r	-0.56	-0.46	-0.45	-0.45
r^2	0.50	0.37	0.36	0.35
a_0	7220.86	15785.93	10351.83	11481.03
a_1	-1388.73	-2220.84	-1573.33	-1671.63
a_2	60.14	70.65	53.21	55.46

4.2.2　新型コロナウイルス感染者数との関係

図 4.15 に都道府県ごとの感染者数を比較した図を示す。都道府県ごとのインフルエンザの感染者数を比較した場合と同様に、首都圏や関西圏で感染者数が集中していることがわかる。図 4.16 に 2020 年 4 月～10 月にかけての感染者数の推移を示す。この図からは、5 月から 6 月にかけては感染者数がほとんど発生していないが、7 月に急増してそこからは減少傾向にあることがわかる。

図 4.17 に主要 4 都市の外気温湿度と新型コロナウイルス感染者数の関係性を示す。この図からは、北海道とその他 3 都市で傾向が違うことが確認できる。北海道の場合は気温が 10℃～20℃絶対湿度 10g/kg 以下で感染者数が集中しているが、他の 3 都市は気温が約 30℃絶対湿度 20g/kg 以上で感染者数が集中していることがわかる。

外気温湿度と感染者数を回帰式で解析した結果を表 4.21 と表 4.22 に示す。北海道のみ r 値が-0.35 で負の相関が確認され、その他 3 都市では r 値が約 0.50 と正の相関を示した。北海道の外気温と感染者数の関係性を示した二次曲線の回帰係数を例にすると、気温が 1℃上昇すると感染者が 0.5 人増加するという予測値になる。同じ条件で大阪を例にすると、気温が 1℃上昇すると感染者が 1.13 人上昇するという予測値になる。二次曲線による感染者数の予測値はどちらも正の値となった。

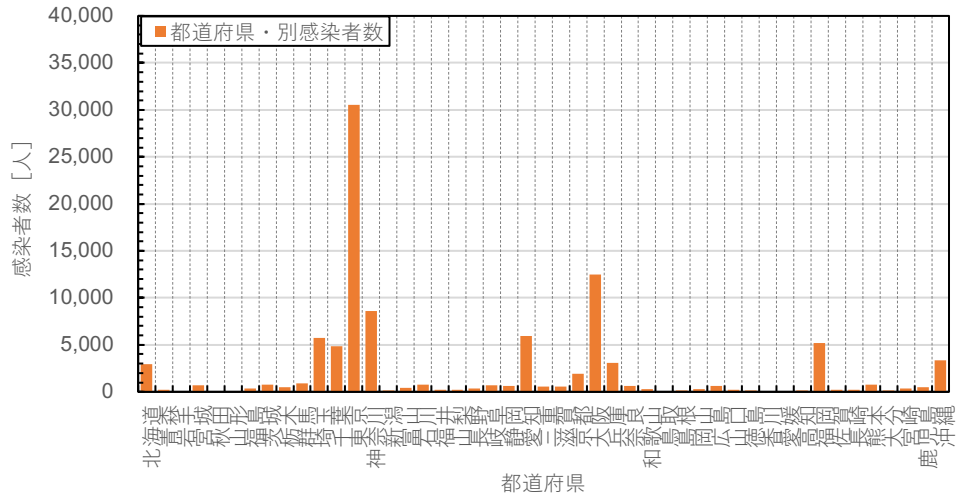


図 4.15　都道府県ごとの感染者数比較

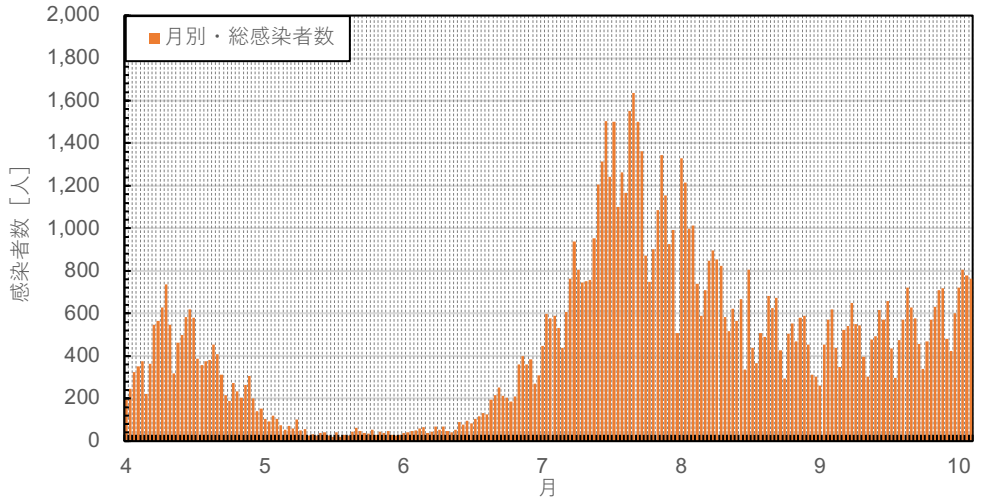


図 4.16　全国の感染者数推移（4 月～10 月）

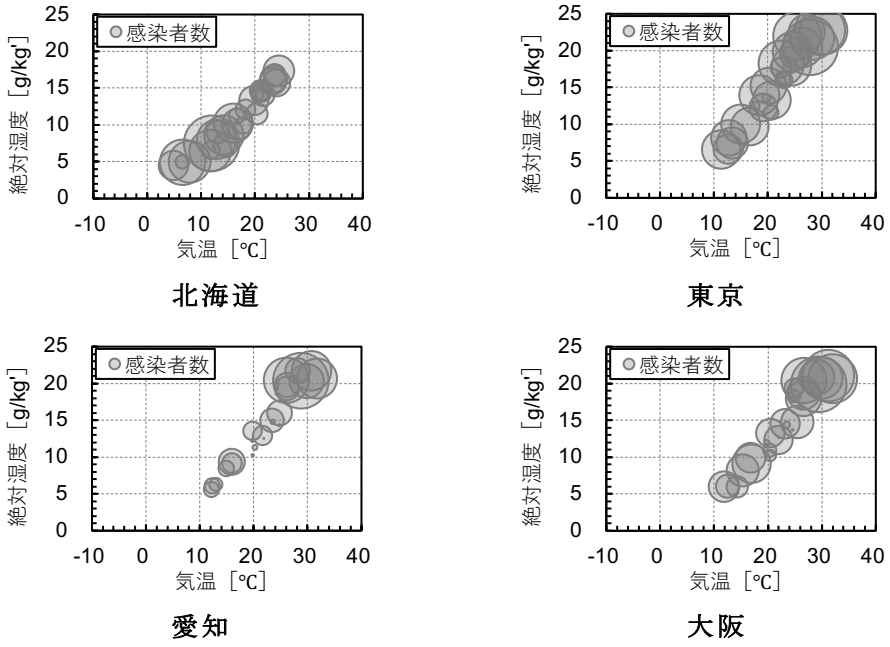


図 4.17 外気温湿度と新型コロナウイルス感染者数の関係(2020 年)

表 4.21　外気温と新型コロナウイルス感染者数の解析結果(2020 年)

	北海道	東京	愛知	大阪
r	−0.35	0.45	0.42	0.46
r^2	0.13	0.21	0.18	0.27
a_0	0.58	0.68	−0.01	1.24
a_1	−0.03	−0.03	−0.01	−0.12
a_2	0.001	0.01	0.01	0.01

表 4.22　絶対湿度と新型コロナウイルス感染者数の解析結果(2020 年)

	北海道	東京	愛知	大阪
r	−0.31	0.57	0.55	0.58
r^2	0.11	0.36	0.36	0.44
a_0	0.60	0.86	0.41	1.03
a_1	−0.05	−0.08	−0.08	−0.17
a_2	0.002	0.005	0.01	0.01

4.3　海外の気温と感染者数の関係

オーストラリアの 10 年間のインフルエンザ陽性率推移を図 4.18 に示す。この図から 7 月から 9 月にかけてインフルエンザの陽性率が 30%～40%まで上昇するという傾向にあることがわかる。

図 4.19 から図 21 にかけて外気温とインフルエンザ陽性率の関係性を示す。ここで扱う外気温は、表 3.3 に示したオーストラリアの主要 5 都市の平均最高気温と平均最低気温の中央値を扱っている。2012 年や 2019 年は外気温が 20℃以下でインフルエンザ陽性率が 30%を超える傾向にあるが、2018 年や 2020 年は外気温が 30℃以上でインフルエンザ陽性率が最大になることがわかる。図 4.22 に 2020 年の外気温と新型コロナウイルス感染者数の関係性を示す。この図からは外気温が 20℃以下で新型コロナウイルス感染者数が 8000 人を超えていることがわかる。

表 4.21 と表 4.22 に外気温とインフルエンザ陽性率の解析結果を示す。ほとんどの年で気温とインフルエンザ陽性率の間に負の相関があり、正の相関がある年の決定係数は低くなるという傾向が得られた。しかし 2020 年は高い正の相関と決定係数 0.56 という結果が出ており、図 4.21 から陽性率の最大値が 10%であることから 2020 年の新型コロナウイルスとインフルエンザウイルスの同時流行が起きているとは考えにくい。他の 10 年に比べて 2020 年のインフルエンザ陽性率が極端に低いことから、オーストラリアでも日本と同様に新型コロナウイルスに対する感染対策がインフルエンザウイルスの感染防止に影響していると考えられる。

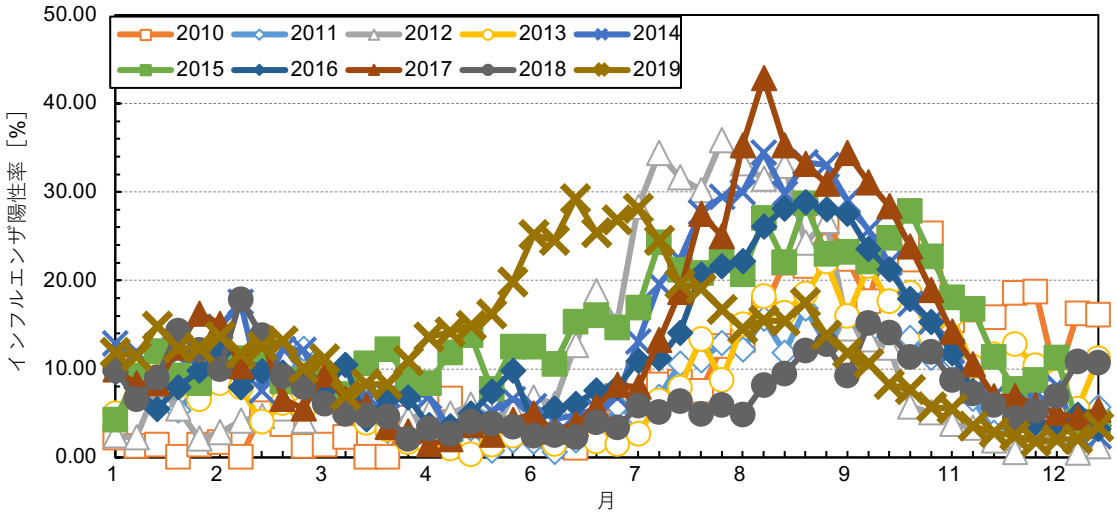


図 4.18　オーストラリアのインフルエンザウイルス陽性率推移（10 年間）

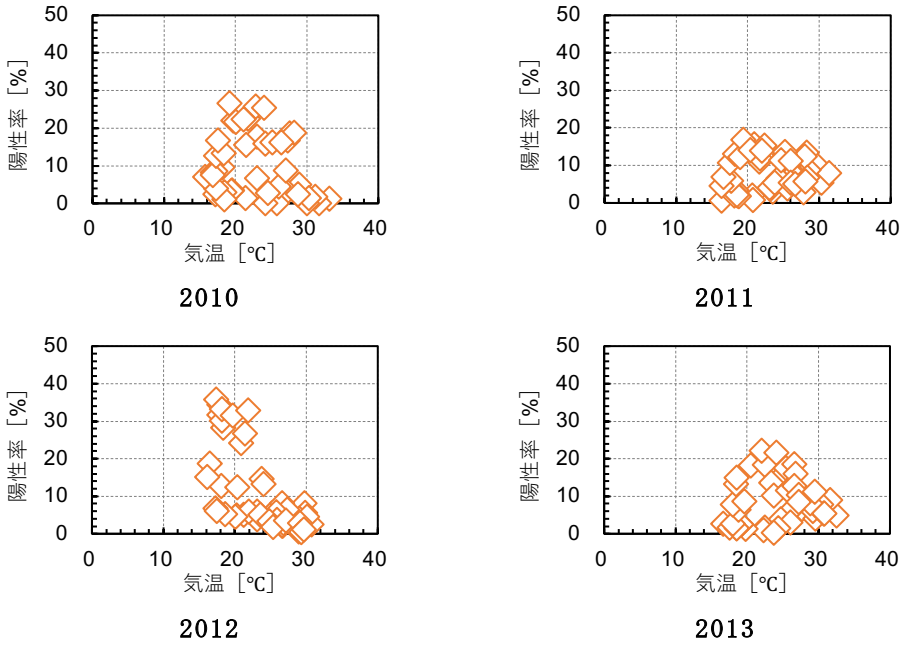


図 4.19 外気温湿度とインフルエンザ陽性率の関係(2010 年～2013 年)

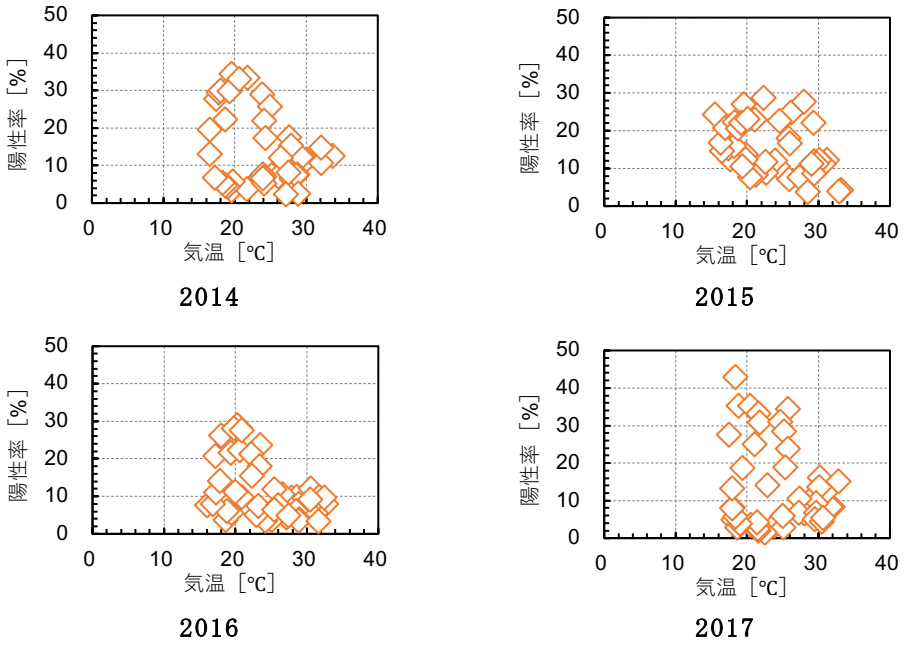


図 4.20 外気温湿度とインフルエンザ陽性率の関係(2014 年～2017 年)

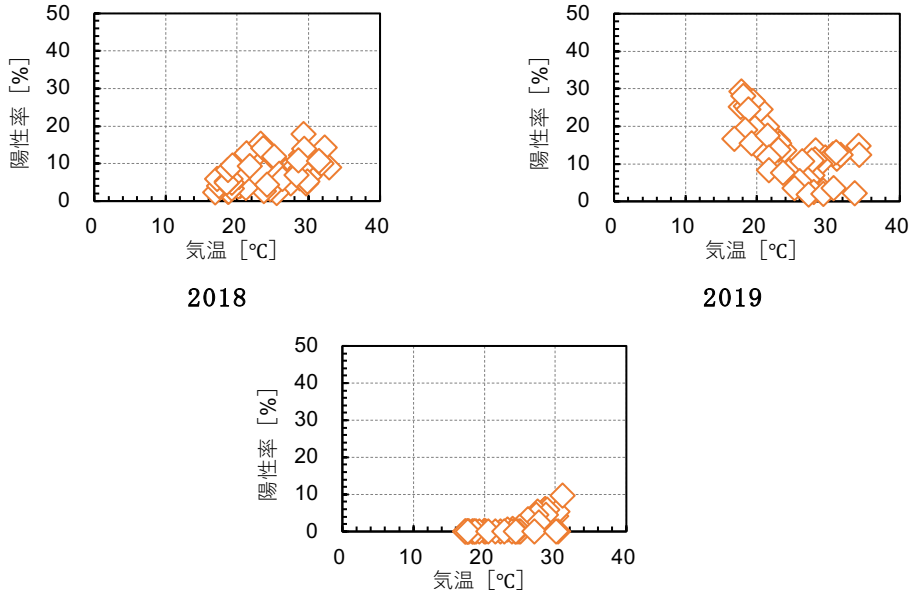


図 4.21 外気温湿度とインフルエンザ陽性率の関係(2018 年～2020 年)

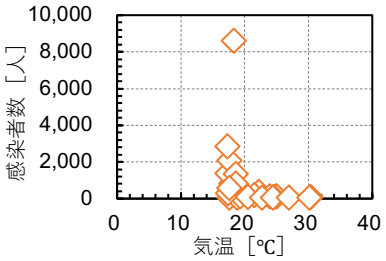


図 4.22 外気温と新型コロナウイルスの関係

表 4.21　外気温とインフルエンザ陽性率の解析結果(2010 年～2015 年)

	2010	2011	2012	2013	2014	2015
r	-0.27	0.14	-0.64	0.10	-0.32	-0.48
r^2	0.22	0.05	0.42	0.15	0.11	0.24
a_0	-65.16	-22.99	82.90	-67.19	52.42	12.28
a_1	7.03	2.56	-4.62	6.28	-2.63	0.89
a_2	-0.15	0.05	0.06	0.12	0.04	-0.03

表 4.22 外気温とインフルエンザ陽性率の解析結果(2016 年～2020 年)

	2016	2017	2018	2019	2020
r	-0.45	-0.23	0.43	-0.66	0.68
r^2	0.26	0.05	0.19	0.63	0.56
a_0	18.87	1.96	-8.34	123.40	18.21
a_1	0.14	1.52	0.94	-8.01	-1.86
a_2	-0.01	-0.04	0.01	0.13	0.04

表 4.23 外気温と新型コロナウイルス感染者数の解析結果(2020 年)

	2020
r	-0.23
r^2	0.05
a_0	442.62
a_1	-10.53
a_2	0.07

4.4　まとめ

日本の新型コロナウイルスの感染者数は、インフルエンザウイルスの感染者数と比較すると少ない。しかしインフルエンザウイルスと違って季節に関係なく夏の高気温高湿度でも感染者が多く発生することから一年中感染に警戒する必要がある。

日本の今年のインフルエンザウイルスの感染者数が少ないことと、オーストラリアの今年のインフルエンザ陽性率が低いことから新型コロナウイルスとインフルエンザウイルスの同時流行は起きにくいと考えられる。

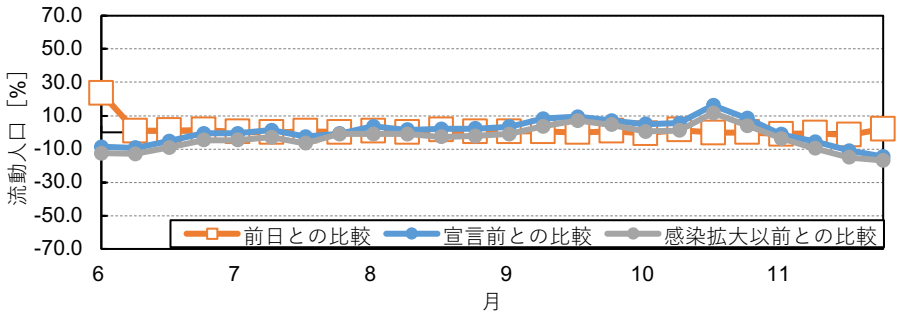
第 5 章　流動人口と新型コロナウイルスの関係性を検討

5.1　都道府県ごとの比較

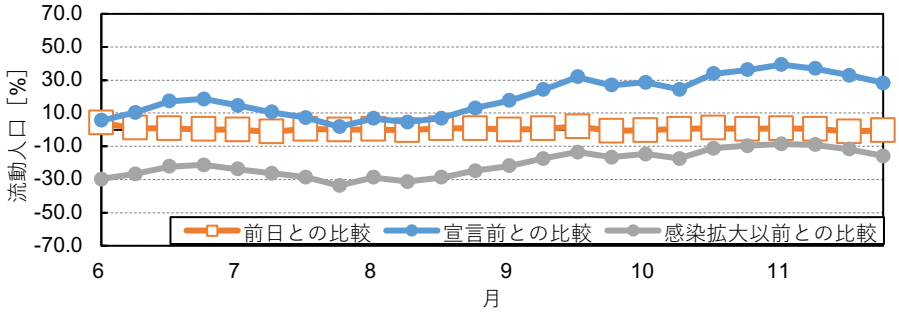
都道府県ごとの流動人口と新型コロナウイルスの感染者数の関係性の比較では、外気温湿度との関係性の分析で対象地域とした主要 4 都市に加えて、宮城，長野，広島，福岡の 4 地域も加えて比較検討をする。図 5.1 と図 5.2 に示すように、流動人口の推移は地域ごとにばらつきがあり、気象条件のように一貫した傾向は無い。

この分析で扱う流動人口はマイナスの値があるが、これは 4 月 7 日という基準値よりも上回るか下回るかという意味である。

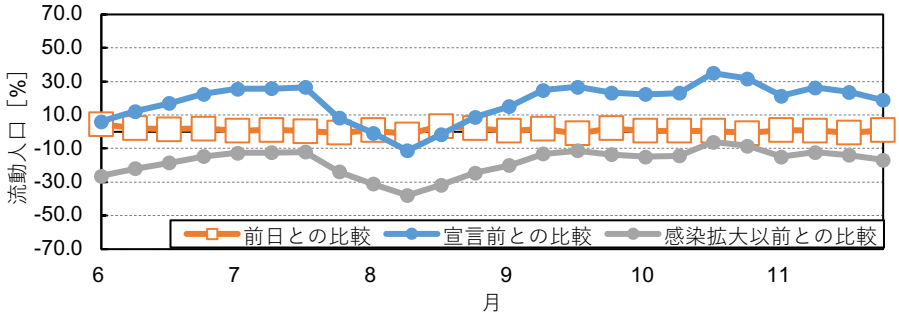
主要 4 都市とその他 4 地域の流動人口と新型コロナウイルス感染者数の関係性を図 5.3 と図 5.4 に示す。地域ごとの回帰式による解析結果を表 5.1 と表 5.2 に示す。8 地域全体で見ると比較的正の関係性が強い。しかし北海道の r 値が 0.09 であることと福岡の r 値が-0.10 であることから、都道府県ごとの分析では流動人口と新型コロナウイルス感染者数の間に有意の関係性があるとは言えないと考える。



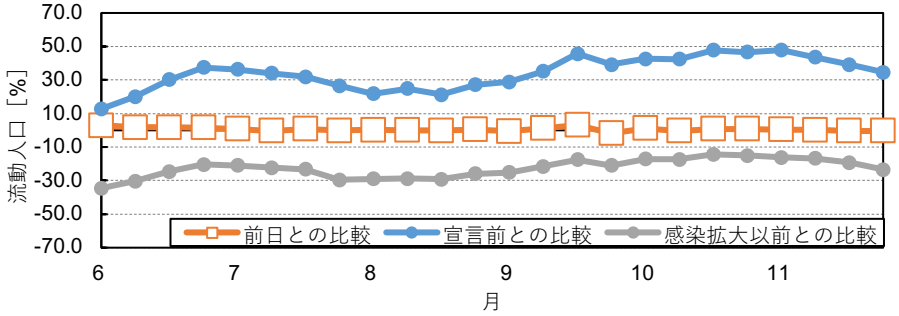
北海道



東京

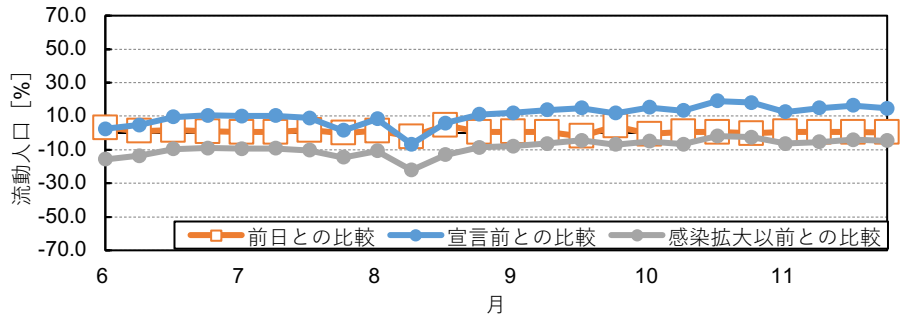


愛知

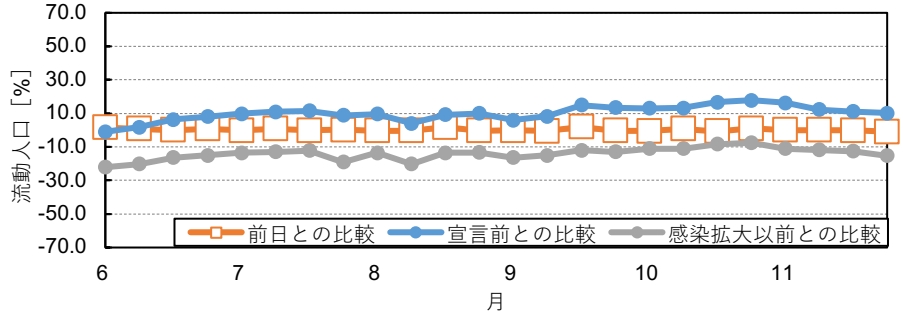


大阪

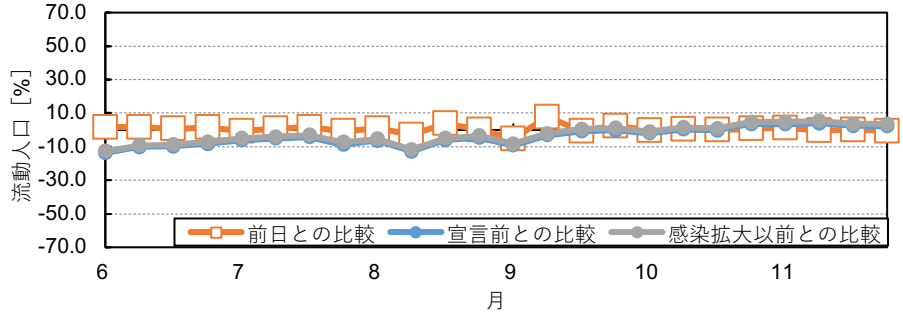
図 5.1 主要 4 都市の流動人口



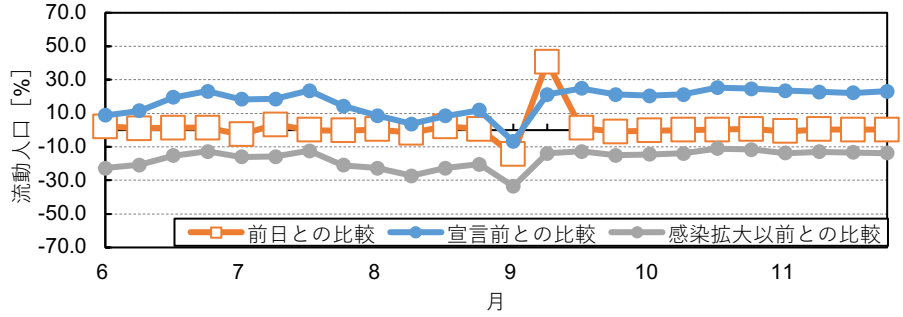
宮城



長野



広島



福岡

図 5.2 その他 4 地域の流動人口

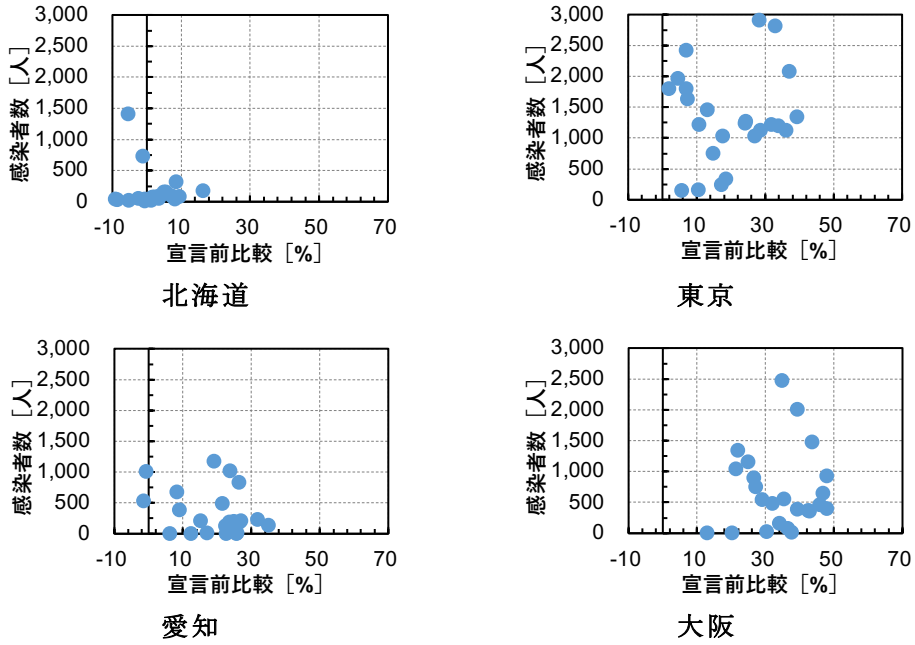


図 5.3 主要 4 都市の流動人口と新型コロナウイルス感染者数の関係

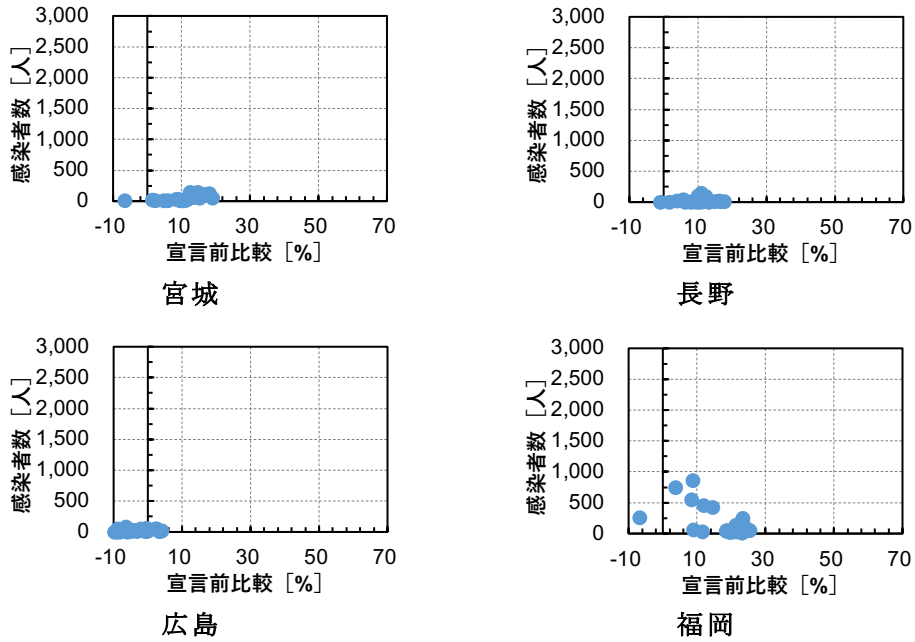


図 5.4 その他 4 地域の流動人口と新型コロナウイルス感染者数の関係

表 5.1 主要 4 都市の流動人口と新型コロナウイルス感染者数の解析結果

	北海道	東京	愛知	大阪
r	0.09	0.33	0.25	0.42
r^2	0.03	0.41	0.16	0.19
a_0	253.34	2058.08	238.57	520.16
a_1	-0.08	-119.35	-11.17	-19.22
a_2	1.71	3.36	0.74	0.70

表 5.2 その他 4 地域の流動人口と新型コロナウイルス感染者数の解析結果

	宮城	長野	広島	福岡
r	0.69	0.43	0.39	-0.10
r^2	0.64	0.27	0.18	0.05
a_0	-5.32	12.81	33.30	220.81
a_1	0.18	-3.39	0.61	12.37
a_2	0.37	0.38	-0.13	-0.65

5.2　全国を視点にした流動人口と感染者数の関係

6 月から 11 月の全国の新型コロナウイルス感染者数推移を図 5.5 に、月毎の流動人口と新型コロナウイルス感染者数の関係性を図 5.6 と図 5.7 に示す。図 5.6 と図 5.7 ではプロットしている点一つ一つが都道府県であり、月合計の感染者数、月平均の流動人口の変化を扱っている。「新規/10 万人」は人口 10 万人あたり新規感染者数のことであり、地域ごとの人口の差が分析結果に反映されないようにこのデータを分析に扱っている。

回帰式による解析結果を表 5.3 に示す。全国で感染者が急増した 7 月は r 値 0.22 で、人の移 動が多い地域ほど感染者数が増加している。感染者数が 減少傾向にある 8 月は r 値-0.07 で人の移動も減少して いる。感染者数が増加傾向にある 9 月～11 月は r 値 0.13 ～0.32 で人の移動と感染者数の正の相関が確認できる。

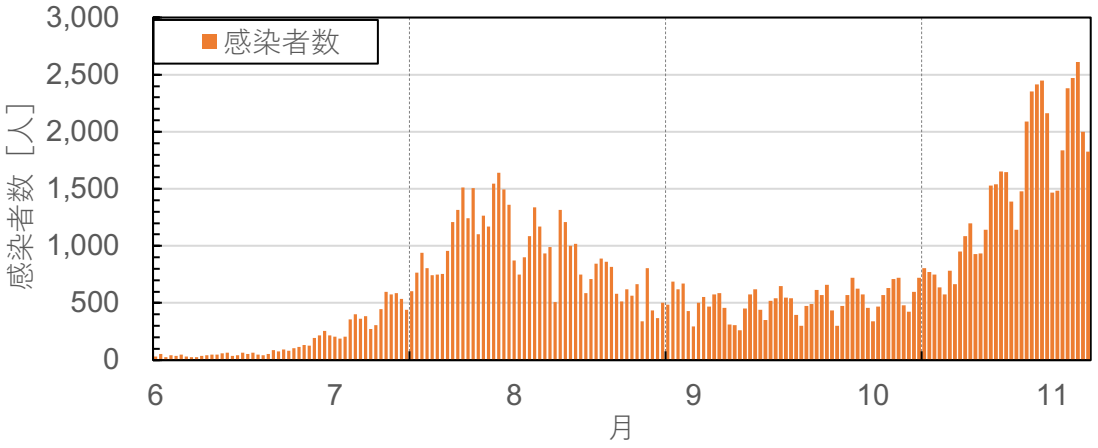
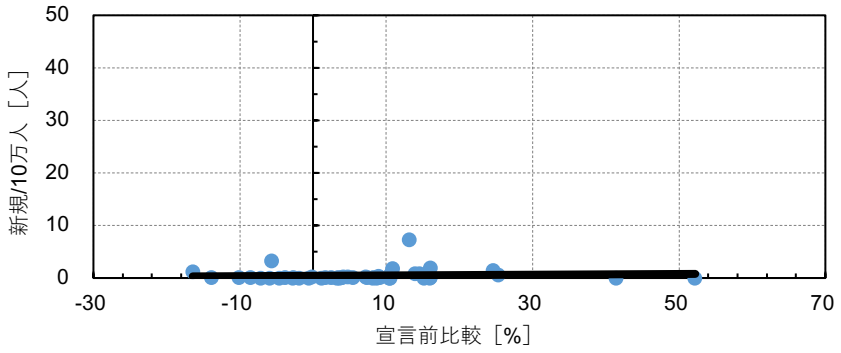
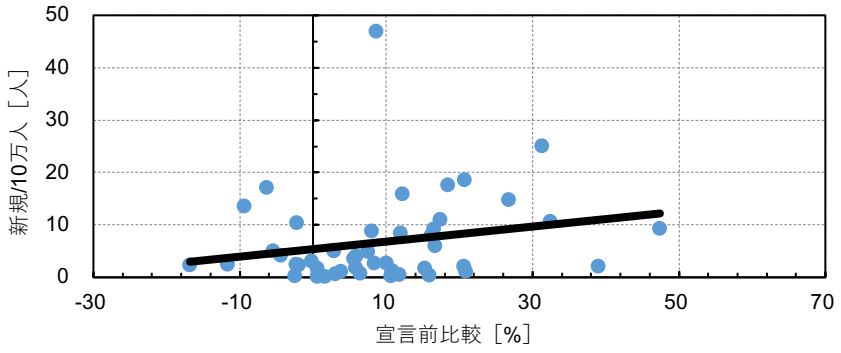


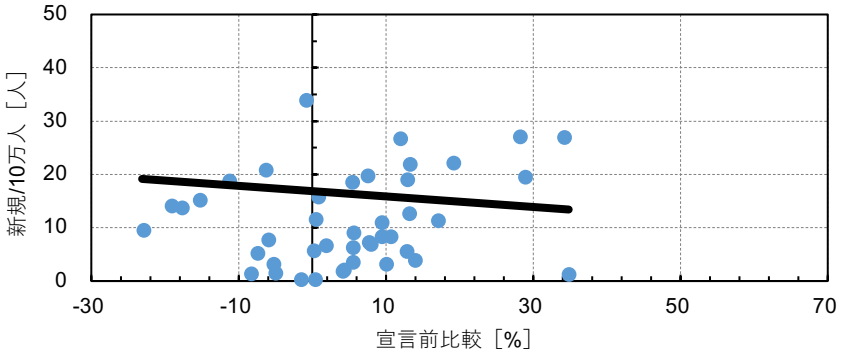
図 5.5　全国の新型コロナウイルス感染者数推移(6 月～11 月)



6 月

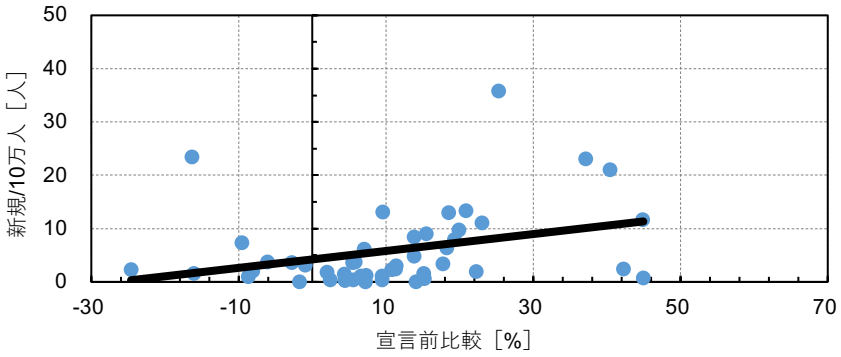


7 月

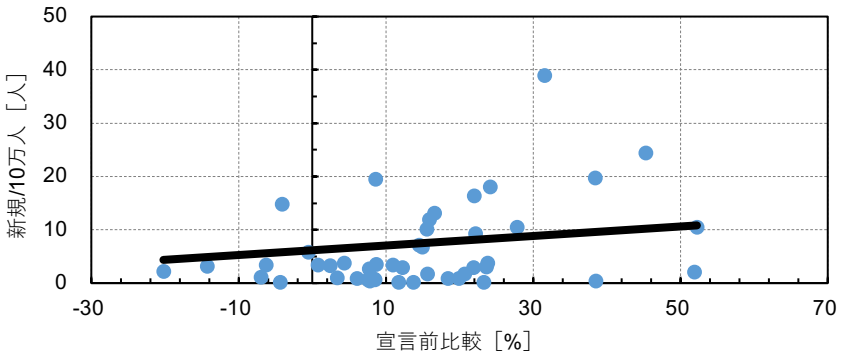


8 月

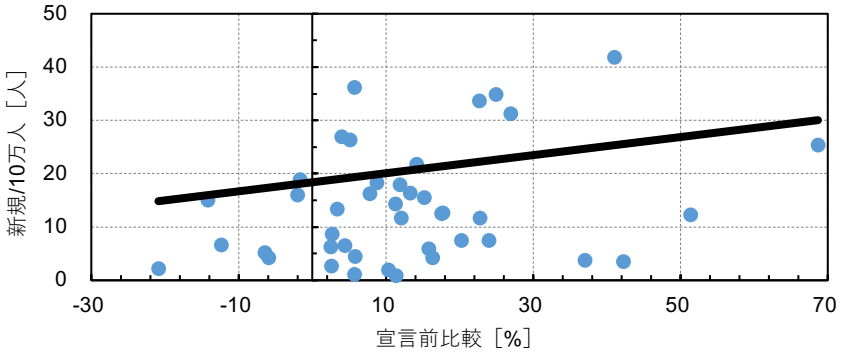
図 5.6 全国の流動人口と感染者数の関係(6 月～8 月)



9 月



10 月



11 月

図 5.7 全国の流動人口と感染者数の関係(9 月～11 月)

表 5.3　全国の流動人口と新型コロナウイルスの解析結果

	6 月	7 月	8 月	9 月	10 月	11 月
r	0.01	0.22	-0.07	0.32	0.13	0.13
r^2	0.01	0.04	0.11	0.14	0.03	0.02
a_0	0.42	5.38	12.86	3.94	6.35	18.74
a_1	0.01	0.13	-0.36	0.05	-0.05	0.04
a_2	-0.01	0.01	0.02	0.01	0.01	0.01

5.3　まとめ

都道府県ごとの分析では、流動人口と新型コロナウイルス感染者数の間に一貫した関係性は確認できなかったが。しかし全国 47 都道府県を対象に感染者数の増減との関係性を分析すると、全国で感染者が増加した月は人の移動も増加、全国で感染者が減少した月は人の移動も減少するという傾向にあることがわかった。

第 6 章　総括

本研究では、外気温や流動人口が呼吸器系感染症の感染者数に及ぼす影響を検討し、以下の知見が得られた。

① 2020 年のインフルエンザウイルス感染者数は、新型コロナウイルス感染対策の影響により減少した。

② 新型コロナウイルスはインフルエンザウイルスと比較して感染者数は少ないが、高気温高湿度で感染者が発生することから年中警戒が必要である。

③ 流動人口の変化が新型コロナウイルス感染者数の変化に影響を及ぼしている。

④ 2020 年のオーストラリアのインフルエンザウイルスの流行は、2019 年と比較して緩やかである。

参考文献

1) David N. Prata, Waldecy Rodrigues, Paulo H. Bermejo:Temperature significantly changes COVID-19 transmittion in (sub) tropical cities of Brazil, Administrasion, University of Brasilia,2020,8

付録

付 1　相関係数と決定係数の計算方法

以下に示す式を用いて、相関係数(r 値)および決定係数(r^2)を算出する。(1)式はピアソンの積率相関係数であり、この式で r 値を求めれば r^2 値はその平方根として求めることができる。しかし感染症の分析が線形では表現しにくいことと、r 値が低くなった場合にピアソンの積率相関係数で r^2 値を求めてしまうとデータの当てはまり具合が高い場合でも低い値となってしまう。このとこを避けるために r 値および r^2 値を別々の計算方法で算出している。(2)式は二次曲線の非線形回帰での r^2 値を求める。

$$r = \frac{S_{xy}}{S_x S_y} \tag{1}$$

$$r^2 = \frac{\Sigma(\hat{y}-\bar{y})^2}{\Sigma(y-\bar{y})^2} \tag{2}$$

付表 1 記号表

S_{xy}:	共分散
S_x:	説明変数xの標準偏差
S_y:	目的変数yの標準偏差
$\hat{y}$:	目的変数yの y 予測値
$\bar{y}$:	目的変数yの平均値

付録

付 2　回帰式プログラム(線形，非線形)

```
import csv
import math
import os
import numpy as np

class Data(): # データクラス
    def __init__(self):
        self.temp_data_list = []
        self.IF_data_list = []

class Average(): # 平均クラス
    def __init__(self):
        self.temp_ave = None
        self.IF_ave = None

    def averageCalc(self):
        temp_total = 0
        IF_total = 0
        for (temp, IF) in zip(data.temp_data_list, data.IF_data_list):
            temp_total += float(temp)
            IF_total += float(IF)
        self.temp_ave = temp_total / len(data.temp_data_list)
        self.IF_ave = IF_total / len(data.IF_data_list)

class Deviation(): # 偏差クラス
    def __init__(self):
        self.temp_deviation_list = []
        self.IF_deviation_list = []

    def deviationCalc(self):
        for (temp, IF) in zip(data.temp_data_list, data.IF_data_list):
            self.temp_deviation_list.append(float(temp)                    -
float(ave.temp_ave))
            self.IF_deviation_list.append(float(IF) - float(ave.IF_ave))

class Dispersion(): # 分散クラス
    def __init__(self):
        self.S_temp2 = None
        self.S_IF2 = None

    def dispersionCalc(self):
```

```
        temp = 0
        IF = 0
        for    (temp_dev,    IF_dev)    in    zip(dev.temp_deviation_list,
dev.IF_deviation_list):
            temp += float(temp_dev) * float(temp_dev)
            IF += float(IF_dev) * float(IF_dev)
        self.S_temp2 = temp / len(dev.temp_deviation_list)
        self.S_IF2 = IF / len(dev.IF_deviation_list)

class standardDeviation(): # 標準偏差クラス
    def __init__(self):
        self.S_temp = None
        self.S_IF = None

    def standardDevationCalc(self):
        self.S_temp = math.sqrt(disp.S_temp2)
        self.S_IF = math.sqrt(disp.S_IF2)

class Covariance(): # 共分散クラス
    def __init__(self):
        self.S_temp_IF = None

    def covarianceCalc(self):
        dev_total = 0
        for    (temp_dev,    IF_dev)    in    zip(dev.temp_deviation_list,
dev.IF_deviation_list):
            dev_total += float(temp_dev) * float(IF_dev)
        self.S_temp_IF = dev_total / len(dev.temp_deviation_list)

class regressionLine(): # 回帰直線クラス
    def __init__(self):
        self.inclination = None
        self.intercept = None

    def inclinationCalc(self):
        inc = float(co.S_temp_IF) / float(disp.S_temp2)
        self.inclination = inc

    def interceptCalc(self):
        itc    =    float(ave.IF_ave)    -    (((float(co.S_temp_IF)    /
float(disp.S_temp2))) * float(ave.temp_ave))
        self.intercept = itc
```

```
    def regressionLinePrint(self, path):
        self.inclination = '{:.4f}'.format(float(self.inclination))
        self.intercept = '{:.4f}'.format(float(self.intercept))
        print(' ')
        print('-----------------------------------')
        print(os.path.basename(path).split('.', 1)[0]) # ファイル名
        print('回帰直線')
        print('y = ' + str(self.inclination) + 'x + ' + str(self.intercept))

class Coefficient(): # 係数クラス
    def __init__(self):
        self.correlation_coef = None # 相関係数
        self.determination_coef = None # 決定係数

    def correlationCalc(self):
        r = float(co.S_temp_IF) / (float(sd.S_temp) * float(sd.S_IF))
        self.correlation_coef = r
        return r

    def determinationCalc(self):
        rr = float(self.correlationCalc()) * float(self.correlationCalc())
        self.determination_coef = rr

    def coefPrint(self):
        self.correlation_coef                                                =
'{:.4f}'.format(float(self.correlation_coef))
        self.determination_coef                                              =
'{:.4f}'.format(float(self.determination_coef))
        print('相関係数')
        print(self.correlation_coef)
        print('決定係数')
        print(self.determination_coef)
        print('-----------------------------------')

##二次曲線回帰クラス
class Exponentiation(): # 冪乗クラス
    def __init__(self):
        self.x_0_sum = None
        self.x_1_sum = None
        self.x_2_sum = None
        self.x_3_sum = None
        self.x_4_sum = None
```

```
def XZeroCalc(self):
    x_zeroList = []
    dataSum = 0
    for dt in data.temp_data_list:
        x_zeroList.append(dt ** (0))
    for i in x_zeroList:
        dataSum += i
    self.x_0_sum = dataSum

def XOneCalc(self):
    x_oneList = []
    dataSum = 0
    for dt in data.temp_data_list:
        x_oneList.append(dt ** (1))
    for i in x_oneList:
        dataSum += i
    self.x_1_sum = dataSum

def XTwoCalc(self):
    x_twoList = []
    dataSum = 0
    for dt in data.temp_data_list:
        x_twoList.append(dt ** (2))
    for i in x_twoList:
        dataSum += i
    self.x_2_sum = dataSum

def XThreeCalc(self):
    x_threeList = []
    dataSum = 0
    for dt in data.temp_data_list:
        x_threeList.append(dt ** (3))
    for i in x_threeList:
        dataSum += i
    self.x_3_sum = dataSum

def XFourCalc(self):
    x_fourList = []
    dataSum = 0
    for dt in data.temp_data_list:
        x_fourList.append(dt ** (4))
    for i in x_fourList:
        dataSum += i
```

```
        self.x_4_sum = dataSum

class XYmultiplication(): # xy を求めるクラス
    def __init__(self):
        self.xy_0_sum = None
        self.xy_1_sum = None
        self.xy_2_sum = None
        self.xy_list = []

    def XYZeroCalc(self):
        x_zeroList = []
        dataSum = 0
        for dt in data.temp_data_list:
            x_zeroList.append(dt ** (0))
        for i, j in zip(x_zeroList, data.IF_data_list):
            dataSum += (i * j)
        self.xy_0_sum = dataSum
        self.xy_list.append(dataSum)

    def XYOneCacl(self):
        x_oneList = []
        dataSum = 0
        for dt in data.temp_data_list:
            x_oneList.append(dt ** (1))
        for i, j in zip(x_oneList, data.IF_data_list):
            dataSum += (i * j)
        self.xy_1_sum = dataSum
        self.xy_list.append(dataSum)

    def XYTwoCacl(self):
        x_twoList = []
        dataSum = 0
        for dt in data.temp_data_list:
            x_twoList.append(dt ** (2))
        for i, j in zip(x_twoList, data.IF_data_list):
            dataSum += (i * j)
        self.xy_2_sum = dataSum
        self.xy_list.append(dataSum)

class InverseMatrix(): # 逆行列の計算クラス
    def __init__(self):
        self.matrix_list = [] # 逆行列の値リスト
```

```
    def MatrixCalc(self):
        a_list = [[ex.x_0_sum, ex.x_1_sum, ex.x_2_sum], [ex.x_1_sum,
ex.x_2_sum, ex.x_3_sum], [ex.x_2_sum, ex.x_3_sum, ex.x_4_sum]]
        inv_a = np.linalg.inv(a_list)
        for i in inv_a:
            self.matrix_list.append(i[0])
            self.matrix_list.append(i[1])
            self.matrix_list.append(i[2])

class CurveCalc(): # 二次曲線の計算クラス
    def __init__(self):
        self.a0 = None
        self.a1 = None
        self.a2 = None

    def curveCalcMethod(self):
        xy_index_list = [1, 1, 1, 2, 2, 2, 3, 3, 3]
        a0_num = 0
        a1_num = 0
        a2_num = 0
        for i, j in zip(range(0, 9), xy_index_list):
            if i in range(0, 7, 3):
                a0_num += (Inverse.matrix_list[i] * multi.xy_list[j-1])
            elif i in range(1, 8, 3):
                a1_num += (Inverse.matrix_list[i] * multi.xy_list[j-1])
            elif i in range(2, 9, 3):
                a2_num += (Inverse.matrix_list[i] * multi.xy_list[j-1])
            else:
                print('エラー')
        self.a0 = a0_num
        self.a1 = a1_num
        self.a2 = a2_num

    def curvePrint(self):
        print('二次曲線')
        a0_new = '{:.4f}'.format(self.a0)
        a1_new = '{:.4f}'.format(self.a1)
        a2_new = '{:.4f}'.format(self.a2)
        print('y = ' + str(a2_new) + ' x^2 + (' + str(a1_new) + ') x + (' +
str(a0_new) + ')')

class CurveCoef(): # 二次曲線の係数計算クラス
    def __init__(self):
```

```
        self.reg_variation = None # 回帰変動の値
        self.resi_fluc = None # 残差変動の値

    def RegressionVariationCalc(self): # 回帰変動
        reg_var_sum = 0
        for x in data.temp_data_list:
            pre_y = (float(curve.a2) * (x ** (2))) + (float(curve.a1) * x)
+ (float(curve.a0))
            reg_var_sum += ((pre_y - ave.IF_ave) ** (2))
        self.reg_variation = reg_var_sum

    def ResidualFluctuation(self): # 残差変動
        resi_sum = 0
        for data_x, data_y in zip(data.temp_data_list, data.IF_data_list):
            pre_y = (float(curve.a2) * (data_x ** (2))) + (float(curve.a1)
* data_x) + (float(curve.a0))
            resi_sum += ((data_y - pre_y) ** (2))
        self.resi_fluc = resi_sum

    def DeterminationCoef(self): # 決定係数
        r2 = (self.reg_variation / (self.reg_variation + self.resi_fluc))
        r2 = '{:.4f}'.format(r2)
        print('決定係数')
        print(r2)
        print('-----------------------------------')
        print(' ')


#csv のパス
path_list = []

for path in path_list:
    data = Data() # データクラスのインスタンス
    ##CSV 読み込み処理
    with open(path, encoding = "utf-8-sig") as f:
        reader = csv.reader(f)
        for row in reader:
            data.temp_data_list.append(float(row[0]))
            data.IF_data_list.append(float(row[1]))

    ##単回帰分析インスタンス
    ave = Average() # 平均計算クラスのインスタンス
    dev = Deviation() # 偏差クラスのインスタンス
```

```
disp = Dispersion() # 分散クラスのインスタンス
sd = standardDeviation() # 標準偏差クラスのインスタンス
co = Covariance() # 共分散クラスのインスタンス
reg = regressionLine() # 回帰直線クラスのインスタンス
coef = Coefficient() # 係数クラスのインスタンス

ave.averageCalc()
dev.deviationCalc()
disp.dispersionCalc()
sd.standardDevationCalc()
co.covarianceCalc()
reg.inclinationCalc()
reg.interceptCalc()
reg.regressionLinePrint(path)
coef.determinationCalc()
coef.coefPrint()


#二次曲線回帰インスタンス
ex = Exponentiation() # 冪乗クラス
multi = XYmultiplication() # xy の掛け算クラス
Inverse = InverseMatrix() # 逆行列の計算クラス
curve = CurveCalc() # 二次曲線の計算クラス
curve_coef = CurveCoef() # 二次曲線の係数クラス

ex.XZeroCalc()
ex.XOneCalc()
ex.XTwoCalc()
ex.XThreeCalc()
ex.XFourCalc()
multi.XYZeroCalc()
multi.XYOneCacl()
multi.XYTwoCacl()
Inverse.MatrixCalc()
curve.curveCalcMethod()
curve.curvePrint()
curve_coef.RegressionVariationCalc()
curve_coef.ResidualFluctuation()
curve_coef.DeterminationCoef()
```